

;login:

The USENIX Association Newsletter

Volume 11, Number 3

May/June 1986

CONTENTS

Results of the 1986 Election of Officers and Directors for USENIX Association	3
Appeal for Papers and Thank You	3
Executive Director's Report	4
1986 Membership Survey Results	5
Request for UUCP and/or Usenet Proposals	10
Publications Available	11
Results of the Second Annual USENIX Go Tournament	12
The Third USENIX Computer Go Tournament and Second Championship	13
4.3BSD Released	14
1986 Summer USENIX Conference at Atlanta	15
HoneyDanBer UUCP – Bringing UNIX Systems into the Information Age	27
<i>Bill Rieken and Jim Webb</i>	
Future Meetings	37
Local User Groups	38

The closing date for submissions for the next issue of ;login: is July 3, 1986



THE PROFESSIONAL AND TECHNICAL UNIX® ASSOCIATION

;login:

NOTICE

;login: is the official newsletter of the USENIX Association, and is sent free of charge to all members of the Association.

The USENIX Association is an organization of AT&T licensees, sub-licensees, and other persons formed for the purpose of exchanging information and ideas about UNIX[†] and similar operating systems and the C programming language. It is a non-profit corporation incorporated under the laws of the State of Delaware. The officers of the Association are:

President	Alan G. Nemeth
Vice-President	Deborah K. Scherrer
Secretary	Lewis A. Law
Treasurer	Waldo M. Wedel
Directors	Thomas Ferrin Steve C. Johnson Lou Katz Michael D. Tilson
Executive Director	James E. Ferguson

The editorial staff of *;login:* is:

Editor and Publisher	James E. Ferguson usenix!jim
Copy Editor	Michelle Peetz {masscomp,usenix}!mmp
Humor Editor	Rob Kolstad {convex,usenix}!kolstad
Production Editor	Tom Strong usenix!tom
Editorial Director	Lou Katz {ucbvax,usenix}!lou

Other staff members are:

Betty J. Madden	Office Manager
Emma Reed	Membership Secretary
Jordan Hayes	Technical Consultant

USENIX Association Office
P.O. Box 7
El Cerrito, CA 94530
(415) 528-8649
{ucbvax,decvax}!usenix!office

Judy DesHarnais Conference Coordinator
USENIX Conference Office
P.O. Box 385
Sunset Beach, CA 90742
(213) 592-3243 or 592-1381
{ucbvax,decvax}!usenix!judy

John L. Donnelly Exhibit Manager

USENIX Exhibit Office
Oak Bay Building
4750 Table Mesa Drive
Boulder, CO 80303
(303) 499-2600
{ucbvax,decvax}!usenix!john

Contributions Solicited

Members of the UNIX community are heartily encouraged to contribute articles and suggestions for *;login:*. Your contributions may be sent to the editors electronically at the addresses above or through the U.S. mail to the Association office. The USENIX Association reserves the right to edit submitted material.

;login: is produced on UNIX systems using *troff* and a variation of the *-me* macros. We appreciate receiving your contributions in *n/troff* input format, using any macro package. If you contribute hardcopy articles please leave left and right margins of 1" and a top margin of 1½" and a bottom margin of 1¼". Hardcopy output from a line printer or most dot-matrix printers is not reproducible.

Acknowledgments

The Association uses a VAX[‡] 11/730 donated by the Digital Equipment Corporation for support of office and membership functions, preparation of *;login:*, and other association activities. It runs 4.2BSD, which was contributed, installed, and is maintained by mt Xinu. The VAX uses a sixteen line VMZ-32 terminal multiplexor donated by Able Computer of Irvine, California.

Connected to the VAX is a QMS Lasergrafix^{*} 800 Printer System donated by Quality Micro Systems of Mobile, Alabama. It is used for general printing and draft production of *;login:* with *ditroff* software provided by mt Xinu.

This newsletter may contain information covered by one or more licenses, copyrights, and/or non-disclosure agreements. No reproduction of this newsletter in its entirety or in part may be made without written permission of the Association.

[‡]VAX is a trademark of Digital Equipment Corporation.

^{*}Lasergrafix is a trademark of Quality Micro Systems.

[†]UNIX is a registered trademark of AT&T.

Results of the 1986 Election of Officers and Directors for USENIX Association

The following members were elected officers and directors:

	<i>Votes Received</i>	<i>Abstentions</i>
Alan Nemeth, President	498	100
Deborah Scherrer, Vice President	527	71
Wally Wedel, Secretary	482	116
Steve Johnson, Treasurer	519	79
Kirk McKusick, Director	453	
David Yost, Director	351	
Rob Kolstad, Director	347	
John Quarterman, Director	331	

The following members were not elected:

Mike O'Dell	306
Mike Tilson	255
Peter Honeyman	242

1 write-in vote for Peter Honeyman for President

Total ballots received = 598

Appeal for Papers and Thank You

We are trying to make ;login: more interesting to the members by including more technical articles. But we need your help to do so. We are appealing to you for your help in submitting to us technical papers which are appropriate for publication in ;login:. The primary standards on which papers will be evaluated for publication are:

1. Interest to a significant number of USENIX members
2. Technically sound

If you have a paper and are willing to share it with your fellow USENIX members, please send it to us - either electronically or hardcopy - at the Association Office.

We thank the following authors whose papers were submitted and published in this and the two prior issues of ;login:

Brian Bershad
Load Balancing With Maitre d'
Erik E. Fair
A Perspective on the USENET
James S. Schoner
Ease: A Configuration Language for Sendmail
Eugene H. Spafford and John C. Flaspohler
A Report on the Accuracy of Some Floating
Math Functions on Selected Computers
Bill Rieken and Jim Webb
HoneyDanBer UUCP - Bringing UNIX®
Systems into the Information Age

Executive Director's Report

Since I have been your executive director for a full year, now seems to be a good time to review USENIX's accomplishments and firsts during the past year. I certainly do not take credit for them. Rather it was through the combined efforts of the board of directors, staff, and members all working together that we have been so successful.

1. Numerically, financially, and certainly attendee evaluations rank the Denver Conference in January as USENIX's best winter conference.
2. USENIX tried and proved that a different format for the winter conferences – concurrent tutorials and symposia – better meets the needs of more members.
3. USENIX proved to itself and the entire UNIX community that it can have a most successful winter conference without being tied to UniForum.
4. The Portland Conference last June – again numerically, financially, and attendee evaluations – is the best summer conference USENIX has had.
5. It will be hard to equal or surpass the 10th anniversary celebration at Alderbrook as part of the Portland Conference.
6. It will even be harder to surpass the excellent quality of the Portland Proceedings that Tektronix prepared.
7. Both the Denver and Portland Conferences confirmed that USENIX tutorials are in great demand and that the tutorials are one of the most valuable and visible services USENIX offers its membership and the UNIX community.
8. *;login:* is gradually improving with each successive issue – better technical papers are now being voluntarily submitted for publication in *;login:*.
9. Two firsts have been scored in *;login:* – last issue was the first time pictures have appeared, and this issue is the first use of color.
10. A simple and effective dues billing system was implemented, and for the first time every member was billed for his or her annual membership dues.
11. Paid membership now exceeds 1,700 and will probably surpass 2,000 after the Atlanta Conference – another first.
12. Financially the association is in great shape with over half a million dollars in reserves.
13. The first comprehensive membership survey was conducted with an astounding 83% return – providing solid information for planning and improving the benefits and services to better meet the members' needs.
14. The survey revealed most members prefer USENIX's winter conferences to be concurrent (same city, same time, but different hotels) with /usr/group's UniForum trade show – thus the next two winter conferences have been scheduled to be concurrent with UniForum in Washington in 1987 and Dallas in 1988.
15. The board of directors is discussing and developing goals, purposes, and directions for the organization covering the next two or three years.
16. For the first time members who aspired to be directors were nominated by petition (rather than being cajoled into serving) and elected to the board of directors.
17. The sale of 4.2BSD manuals far exceeded expectations, and USENIX is underwriting the cost of improving the indexing system for the 4.3BSD manuals.
18. USENIX now has an executive director, staff, and three offices – the main office in El Cerrito, the conference office in Sunset Beach in California, and the exhibits and tutorials office in Boulder, Colorado.
19. USENIX purchased an ISI computer and acquired the IEEE meeting registration software for conference registrations – now USENIX is using UNIX[®] at its conferences, rather than an IBM XT.
20. We have a new slogan which quickly identifies and describes us: **USENIX – THE PROFESSIONAL AND TECHNICAL UNIX[®] ASSOCIATION.**

Thank you for your cooperation and support during the past year. Hopefully we will be even more successful this year.

Jim Ferguson
Executive Director

1986 Membership Survey Results

Jim Ferguson

According to the results of the 1986 Membership Survey the *mythically* average or typical USENIX member:

- is employed by a large corporation
- is a manager or programmer with management responsibilities
- has a master's degree
- is in his mid-thirties
- earns about \$40,000 a year
- has been using UNIX[®] for 5 years
- uses UNIX[®] for:
 - document preparation
 - electronic mail
 - programming and debugging
 - software design and development
- has been a USENIX member for 3 years
- is also a member of ACM and IEEE
- has attended at least one USENIX Conference
- thinks USENIX Conferences and ;login: should remain the same, covering both highly technical topics and general interest topics
- thinks USENIX and /usr/group should hold concurrent (same city, same time) winter meetings
- reads:
 - ;login:
 - Byte
 - Communications of the ACM
 - IEEE Computer Magazine
 - UNIX[®] Review
 - UNIX[®] /World

That sure shoots down the stereotype image many people have of USENIX members!!!

The returns on this survey far exceeded our wild-hopes. Of the more than 1,700 current dues-paid members, slightly over 500 are new members in 1986, and very few of them participated in the survey.

Survey forms were mailed to only those members who were billed renewal dues for 1986. About 1,000 of the 1,200 who paid their renewal dues for 1986 also answered the survey – an exceptionally high return of 83%!!!

We thank all of you who answered the questions and returned the survey.

As with most surveys when the results are tallied, the author wishes he had asked some of the questions just a little differently. Thus, a few notes and comments are needed to better understand the results.

70-80% of the 332 respondents working for organizations with over 10,000 employees or students are from universities and government agencies. Most of the members do not work for corporations with over 10,000 employees.

The few members who report using UNIX[®] for over 12 years appear to be employees of Bell Laboratories and probably helped develop the system, rather than just exaggerating their experience.

At least 150 members – who indicated they have *NOT* attended any USENIX Conferences or UniForum Shows – favor having joint events thereby distorting the comparative percentages in Question 12. Persons who attended both the 1984 joint meeting in Washington and the 1985 concurrent meetings in Dallas, appear to favor the concurrent meetings format by a wide margin.

The ideas, comments, and suggestions in response to Questions 11, 14, 15, and 16 need further study and consolidation. They will be the subject of a future article in ;login:.

The number of responses to the \$90,000+ income range is too high as several jokesters checked that category even though their experience, education, and positions obviously place them at much lower levels.

There are two surprises. First is the decline both in actual numbers and percentage of members who work for educational institutions. The other surprise is the high percentage of members who are in executive, management, and supervisory positions – over 64%.

The following pages are copies of the survey showing the relevant percentage comparisons as well as the raw data tallies for each question.

1986 USENIX Membership Survey Percentages

Based upon 1,016 survey forms returned by May 7, 1986. Includes multiple answers where appropriate.

1. Employer or organization?

49% Business or Corporation
28% Educational Institution
4% Government Agency
3% Nonprofit Organization
10% Research Institution
5% Self-employed
1% Other

1,111 responses

2. Size of employer or organization?

33% Over 10,000 employees or students
10% Over 5,000 employees or students
15% Over 1,000 employees or student
7% Over 500 employees or students
12% Over 100 employees or students
5% Over 50 employees or students
9% Over 10 employees or students
9% Under 10 employees or students

1,011 responses

3. Your position level or title?

33% Executive or Manager
29% Project Leader or Senior Programmer
20% Programmer or Technical Staff Member
1% Sales or Service Staff Member
2% Dean or Department Chairman
7% Professor - Associate - Assistant
2% Instructor or Teacher
1% Graduate Student or Student
5% Consultant or Self-employed

1,062 responses

4. Education level?

18% Doctorate
7% Work on Doctorate
27% Master's Degree
12% Work on Master's
26% Bachelor's Degree
9% College - No Degree
1% High School Graduate

1,031 responses

5. Number of years USENIX member?

1-22% 2-23% 3-24% 4-13% 5-8%
6-3% 7-2% 8-2% 9-1% 10-2%

948 responses

6. Member of what other organizations?

49% ACM 15% Other
36% IEEE 26% No answer
28% /usr/group

1,016 responses

7. Number of years using UNIX®?

1-3% 2-9% 3-12% 4-10% 5-15%
6-10% 7-7% 8-6% 9-4% 10-10%

11-2% 12-1% 14-.8% 15-.2%

No answer-10% 1,016 responses

8. At its conferences and in ;login: should USENIX:

14% Broaden the scope of coverage to more
general interest topics (e.g., e-mail)
10% Narrow the scope of coverage to more
highly technical topics (e.g., kernel)
76% Remain the same, covering some of both

941 responses

9. Did you attend:

. . . the *concurrent* USENIX Winter Conference and
/usr/group UniForum in Dallas in January 1985?

34% Yes 66% No

. . . the *joint* /usr/group and USENIX UniForum
Conference in Washington, DC, in January 1984?

35% Yes 65% No

. . . the *separate* USENIX Summer Conferences in
Portland in June 1985 and/or Salt Lake City in June
1984?

27% Yes, Portland 25% Yes, Salt Lake City
73% No, Portland 75% No, Salt Lake City

1,016 responses

10. Were you satisfied with the format and arrangements in:

. . . Dallas? 81% Yes 19% No

. . . Washington, DC? 76% Yes 24% No

. . . Portland? 95% Yes 5% No

. . . Salt Lake City? 92% Yes 8% No

11. What improvements do you recommend, or what was unsatisfactory?

(Responses will be in a future issue of ;login:.)

12. Do you think USENIX and /usr/group should again try to hold the USENIX Winter Conference and the /usr/group UniForum either jointly or concurrently in the same city?

12% No – different cities, different times
 32% Yes – concurrent – same city, same time
 31% Yes – joint events
 25% No answer

1,066 responses

13. Earlier USENIX Conferences you attended?

24% Toronto – 7/83 24% San Diego – 1/83
 18% Boston – 7/82 15% Santa Monica – 1/82
 6% Austin – 6/81 12% San Francisco – 1/81
 4% Newark – 6/80 7% Boulder – 1/80
 55% No answer 1,016 responses

14. What areas or topics would you like to see covered at USENIX Conferences?

(Responses will be in a future issue of ;login:.)

15. What areas or topics would you like to see written about in ;login:?

(Responses will be in a future issue of ;login:.)

16. What other benefits and services would you like to see USENIX offer its members?

(Responses will be in a future issue of ;login:.)

17. How are you using UNIX®?

13% Accounting or budgeting
 21% Administration or management
 31% Data base management
 62% Document preparation
 65% Electronic mail
 12% Hardware design and development
 4% Hardware marketing
 71% Programming and debugging
 45% Research
 68% Software design and development
 9% Software marketing
 37% System administration
 29% Teaching or student
 5% Other 1,016 responses

18. Age range?

.2% 15-19 14% 40-44
 4% 20-24 6% 45-49
 21% 25-29 3% 50-54
 26% 30-34 2% 55-59
 23% 35-39 .8% 60+

959 responses

19. Income range?

1% \$00-10,000	15% \$50-60,000
4% \$10-20,000	7% \$60-70,000
12% \$20-30,000	2% \$70-80,000
25% \$30-40,000	2% \$80-90,000
26% \$40-50,000	6% \$90,000 +

808 responses

20. What computer magazines and technical publications do you read?

79% ;login:

1% Access
 6% Business Computer Systems
 51% Byte
 56% Communications of the ACM
 26% CommUNIXations
 7% Computer & Software News
 5% Computer Business News
 20% Computer Design
 9% Computer Graphics World
 27% Computerworld
 1% Data Management Magazine
 29% Datamation
 26% Digital Review
 17% Dr. Dobbs Journal
 9% Electronic Week
 9% Electronic News
 24% Hardcopy
 38% IEEE Computer Magazine
 9% Infosystems
 17% Infoworld
 2% Interface Age
 8% MIS Week
 26% Mini Micro Systems
 13% PC World
 12% PC Tech Journal
 4% Personal Computing
 2% Popular Computing
 1% Programmers Journal
 6% Software News
 14% Software Practice & Experience
 7% Systems Software
 15% The C Journal
 57% UNIX® Review
 52% UNIX® / World
 20% Other
 4% No answer

1,016 responses

1986 USENIX Membership Survey Response Tallies

Based upon 1,016 survey forms returned by May 7, 1986. Includes multiple answers where appropriate.

1. Employer or organization?

538 Business or Corporation
 309 Educational Institution
 47 Government Agency
 38 Nonprofit Organization
 113 Research Institution
 56 Self-employed
 10 Other
 1,111 responses

2. Size of employer or organization?

332 Over 10,000 employees or students
 103 Over 5,000 employees or students
 154 Over 1,000 employees or student
 70 Over 500 employees or students
 121 Over 100 employees or students
 51 Over 50 employees or students
 91 Over 10 employees or students
 89 Under 10 employees or students
 1,011 responses

3. Your position level or title?

345 Executive or Manager
 306 Project Leader or Senior Programmer
 213 Programmer or Technical Staff Member
 12 Sales or Service Staff Member
 23 Dean or Department Chairman
 76 Professor - Associate - Assistant
 17 Instructor or Teacher
 15 Graduate Student or Student
 55 Consultant or Self-employed
 1,062 responses

4. Education level?

185 Doctorate
 75 Work on Doctorate
 280 Master's Degree
 123 Work on Master's
 263 Bachelor's Degree
 94 College - No Degree
 11 High School Graduate
 1,031 responses

5. Number of years USENIX member?

1-204 2-219 3-234 4-122 5-81
 6-34 7-17 8-14 9-5 10-18
 948 responses

6. Member of what other organizations?

501 ACM 157 Other
 370 IEEE 263 No answer
 287 /usr/group
 1,016 responses

7. Number of years using UNIX®?

1-27 2-85 3-124 4-104 5-149
 6-105 7-66 8-62 9-41 10-102
 11-22 12-13 14-9 15-3
 104 No answer 1,016 responses

8. At its conferences and in ;login: should USENIX:

127 Broaden the scope of coverage to more
 general interest topics (e.g., e-mail)
 97 Narrow the scope of coverage to more
 highly technical topics (e.g., kernel)
 717 Remain the same, covering some of both
 941 responses

9. Did you attend:

... the *concurrent* USENIX Winter Conference and
 /usr/group UniForum in Dallas in January 1985?
 346 Yes 670 No
 ... the *joint* /usr/group and USENIX UniForum
 Conference in Washington, DC, in January 1984?
 358 Yes 658 No
 ... the *separate* USENIX Summer Conferences in
 Portland in June 1985 and/or Salt Lake City in June
 1984?
 278 Yes, Portland 257 Yes, Salt Lake City
 738 No, Portland 759 No, Salt Lake City
 1,016 responses

10. Were you satisfied with the format and arrangements in:

... Dallas? 275 Yes 66 No
 ... Washington, DC? 266 Yes 82 No
 ... Portland? 264 Yes 14 No
 ... Salt Lake City? 237 Yes 20 No

11. What improvements do you recommend, or what was unsatisfactory?

(Responses will be in a future issue of ;login:.)

;login:

12. Do you think USENIX and /usr/group should again try to hold the USENIX Winter Conference and the /usr/group UniForum either jointly or concurrently in the same city?

128 No – different cities, different times
344 Yes – concurrent – same city, same time
326 Yes – joint events
268 No answer

1,066 responses

13. Earlier USENIX Conferences you attended?

244 Toronto – 7/83 240 San Diego – 1/83
185 Boston – 7/82 153 Santa Monica – 1/82
65 Austin – 6/81 117 San Francisco – 1/81
38 Newark – 6/80 72 Boulder – 1/80
555 No answer 1,016 responses

14. What areas or topics would you like to see covered at USENIX Conferences?

(Responses will be in a future issue of ;login:.)

15. What areas or topics would you like to see written about in ;login:?

(Responses will be in a future issue of ;login:.)

16. What other benefits and services would you like to see USENIX offer its members?

(Responses will be in a future issue of ;login:.)

17. How are you using UNIX®?

133 Accounting or budgeting
216 Administration or management
317 Data base management
626 Document preparation
658 Electronic mail
120 Hardware design and development
36 Hardware marketing
718 Programming and debugging
461 Research
678 Software design and development
91 Software marketing
372 System administration
298 Teaching or student
46 Other 1,016 responses

18. Age range?

2 15-19 134 40-44
42 20-24 58 45-49
201 25-29 30 50-54
249 30-34 15 55-59
221 35-39 7 60+

959 responses

19. Income range?

12 \$00-10,000	125 \$50-60,000
34 \$10-20,000	57 \$60-70,000
95 \$20-30,000	20 \$70-80,000
200 \$30-40,000	12 \$80-90,000
209 \$40-50,000	44 \$90,000 +

808 responses

20. What computer magazines and technical publications do you read?

804 ;login:

12 Access
58 Business Computer Systems
517 Byte
566 Communications of the ACM
266 CommUNIXations
74 Computer & Software News
53 Computer Business News
205 Computer Design
93 Computer Graphics World
274 Computerworld
12 Data Management Magazine
298 Datamation
261 Digital Review
172 Dr. Dobbs Journal
94 Electronic Week
96 Electronic News
248 Hardcopy
381 IEEE Computer Magazine
88 Infosystems
176 Infoworld
17 Interface Age
83 MIS Week
266 Mini Micro Systems
130 PC World
126 PC Tech Journal
41 Personal Computing
21 Popular Computing
15 Programmers Journal
64 Software News
144 Software Practice & Experience
73 Systems Software
157 The C Journal
579 UNIX® Review
526 UNIX®/World
205 Other
39 No answer

1,016 responses

Request for UUCP and/or Usenet Proposals

The combination UUCP mail service and Usenet news service has proved to be very attractive and useful to our community. However, there are a number of problems with the current set of services, due to technical, organizational, and financial problems. Some problems include:

- Service is erratic and unreliable.
- Security is virtually non-existent.
- Efficiency is low. Routing of mail and news does not minimize cost or maximize reliability.
- Overall cost is extremely high, in terms of telecommunications charges, CPU utilization, and disk utilization, especially when the total over all sites is considered. Many of the costs are currently "hidden."
- Costs are not allocated fairly. Continued service depends on the continued goodwill of voluntary "backbone" sites who absorb many of the costs. Usenet (the primary cost generator) is in fact very vulnerable to a chain reaction – if a few key sites drop out, the load and cost at the remaining sites increases dramatically, leading to more dropouts, and so on.
- Traffic is growing without bound, with no sign of abating.
- There are few standards or too many (conflicting) standards.
- There is no established uniform way of addressing messages.
- The "useful information" content of netnews appears low.

As the network grows these problems seem to be getting worse, and many fear the ultimate demise of a useful tool.

The USENIX Association has funded some efforts in aid of UUCP/Usenet. These include the network mapping project and the Stargate experiment.

The Association would now like to receive proposals by which we could fund or aid projects that would solve some or most of the UUCP/Usenet network problems. At this point, we do not have any set agenda or preconceived notions; we are open to any reasonable proposal.

We would like proposals that answer the following questions:

- What user needs are met? What user problems are solved? How was this determined?
- Who is going to take action, and how are they organized? What person(s) or group(s) are going to take the lead in making things happen? If needed, who will take care of any proposed on-going operations?
- How much does it cost? What funding is required from USENIX? Will USENIX recover any of its expenditures? If needed, where will on-going operating funds come from?
- What technical solutions are proposed? What technology exists, and what must be developed?

The order of the above is significant; we are most interested in a good analysis of user needs and problems, along with an understanding of who exactly will meet these needs. We want to know: "What functions are to be accomplished, and who will lead the charge?"

Next we want to know how much money is wanted, and whether any solutions seem technically feasible. However, we believe these issues are much easier to deal with if the first questions have good answers.

This request for proposals is not a formal bid. The USENIX Association has not yet allocated any funds for this purpose, nor has it made a commitment that it will indeed accept any proposal. However, we do feel that modest funds could be made available for one or a few deserving projects. The Association feels it could fund projects on the order of \$10,000 if justified. Larger numbers would almost certainly require some cost recovery. As an upper bound, we would consider \$50,000 to be financially out of the question without an assurance of some cost recovery in the near future. The Association realizes that the amount of money it can make available is small relative to the millions of dollars being spent on UUCP/Usenet. However, we feel we can play a role in planting the seeds of a solution.

The Association is willing to consider a wide variety of projects, including research, seed capital for an operating organization, studies intended to attract interest from telecommunications carriers, or small projects that directly solve at least a few problems.

;login:

We would like to be able to discuss proposals at the next USENIX Board meeting in June. If a proposal can not be prepared by then, we would still be interested in a one page outline. Proposals should be sent to:

UUCP/Usenet Proposals
USENIX Association
P.O. Box 7
El Cerrito, CA 94530

Proposals can also be sent electronically to *usenix!jim*.

Michael Tilson

on behalf of the Board of Directors
of the USENIX Association.

{decvax,utzoo,usenix)!hcr!hcradm!mike

Publications Available

The following publications are available from the Association Office or the source indicated. Prices and overseas postage charges are per copy. California

residents please add applicable sales tax. Payments **must** be enclosed with the order and **must** be in US dollars payable on a US bank.

USENIX Conference Proceedings

Meeting	Location	Date	Price	Overseas Mail		Source
				Air	Surface	
USENIX	Denver	Winter '86	\$20	\$25	\$5	USENIX
USENIX	Portland	Summer '85	\$25	\$25	\$5	USENIX
USENIX	Dallas	Winter '85	\$20	\$25	\$5	USENIX
USENIX	Salt Lake	Summer '84	\$25	\$25	\$5	USENIX
UniForum	Wash. DC	Winter '84	\$30	\$20		/usr/group

EUUG Publications

The following EUUG publications may be ordered from the USENIX Association office.

The EUUG Newsletter, which is published four times a year, is available for \$4 per copy or \$16

for a full-year subscription. The earliest issue available is Volume 3, Number 4 (Winter 1983).

The July 1983 edition of the EUUG Micros Catalog is available for \$8 per copy.

Results of the Second Annual USENIX Go Tournament

Peter Langston

The second USENIX Computer Go Tournament was held June 12 at the Summer 1985 USENIX meeting in Portland, Oregon. This is the second full-board computer go competition ever to be held (the first being last year's tournament). There were five programs entered: *goanna*, *gorilla*, *nemesis*, *ogo*, and *oog*. Each entrant played every other entrant twice, once with white and once with black.

Goanna and *gorilla* were written by Bruce Ellis. They employed a pattern-matching strategy that required an astonishingly small amount of cpu time; the game between these two programs was true speed go. *Gorilla* was designed to play a more cutthroat game in which killing opposition stones was a paramount goal.

Nemesis was written by Bruce Wilcox and is available commercially for a variety of microcomputers. It entered the tournament as the reigning champion and clear favorite, having demolished the opposition last year and having played well in a recent tournament with humans. Although judged to be the strongest entrant, it seemed to have trouble with the tense tournament atmosphere and crashed frequently (in 6 out of 8 games), thereby threatening its domination of the standings.

Ogo was written by Peter Langston. It plays a peculiarly thoughtless strategy in which it mirrors its opponent's moves whenever possible. This is not unheard of, even in professional play. Unfortunately, some last minute "enhancements" caused *ogo* to commit suicide with rhythmic regularity. *Oog* was also written by Mr. Langston (who happened to be the tournament organizer); it actually tried to play go (a novel approach for that author).

It was far more difficult to judge the results in this tournament than in an ordinary tournament. Should a loss due to a blatantly illegal move be comparable to a loss from falling into a complete catatonic

stupor and forgetting to move at all? How should either compare to uttering "out of buffers" before discretely resigning? The judges finally decided to pay most attention to genuine go competence. The results were then:

Finish	Entrant	Won	Lost	Other
1	<i>oog</i>	5	2	1
2	<i>nemesis</i>	2	0	6
3	<i>goanna</i>	0	2	6
4	<i>gorilla</i>	0	2	6
5	<i>ogo</i>	0	1	7

Here "Other" refers to a variety of unfortunate outcomes which were scored by a zany scheme concocted by the judges to reflect their notions of how each outcome reflected go skill or lack thereof.

Thus *oog* is the champion, coming from complete obscurity to snatch the title from the favorite. Both of *oog*'s losses were to *nemesis*; one of these games actually had a strong resemblance to a go game. The other genuinely interesting game was a practice game between *nemesis* and itself; it was a close contest but finally black gave up the ghost and fell into a silent torpor.

The game of go is difficult to play well. A sense of broad strategical issues is very important; apparently none of the programs here used the familiar tree search techniques so popular for handling tactical situations. It seems that *nemesis* plays at about the level of 20 kyu judging both from its performance here and its performance in an earlier tournament in which its opponents were human. This is substantially above the level of a beginning player but still very, very far away from being able to beat players drastically weaker than the program's author. Perhaps next year's tournament will bring forth new and stronger programs to astound the USENIX go community, but the champ doesn't seem worried, "Hah! Oog'll cream those wimps!"

The Third USENIX Computer Go Tournament and Second Championship

Peter Langston

	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P	Q	R	S	T	
19	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	19
18	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	18
17	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	17
16	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	16
15	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	15
14	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	14
13	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	13
12	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	12
11	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	11
10	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	10
9	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	9
8	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	8
7	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	7
6	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	6
5	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	5
4	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	4
3	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	3
2	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2
1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P	Q	R	S	T	

Peter Langston
Bell Communications Research
MRE 2E-338
435 South Street
Morristown, NJ 07960

USENIX Computer Go Tournament Rules

Revised April, 1986

Peter Langston

- **A full size board will be used.** The board will be 19 x 19 with columns labeled "A" through "T" (excluding "I") left to right, and rows labeled "19" through "1" top to bottom.

- **Komi will be 5.5 points.** The second player gets a 5.5 point bonus.

- **There will be a time limit.** Each program will be limited to a total of 60 minutes of accumulated "user" time. If a program goes over the time limit it will only be allowed 10 seconds of "user" time for each move (*byo-romi*). If a program goes over the time limit and uses more than 10 seconds of "user" time for a move it will immediately forfeit the game.

- **The programs must not be idle unnecessarily.** If 10 minutes of "real" time elapse with no increase in the current program's "user" time, it will be assumed that the program is stuck and the program will forfeit. (This rule is included to handle cases where a program loses synchronization or is doing something like: "for (;) read(0, buf, sizeof buf);")

- **There will be no forking (around).** Each program must be a single process and must not fork other processes. Forking interferes with the timing mechanism and, like any attempt to evade or fool the timing, will result in a forfeit.

- **A "referee" program will be used.** The tournament will use a "referee" program to execute each competing pair of programs. There will be no command-line arguments, i.e. *argc* will be 1. All communication with the programs will be via the standard input and standard output. Thus the programs must understand a specific set of commands and generate output of a specific form.

1. All input commands to the competing programs will be in the form of lines of text appearing on the standard input and terminated by a newline.

- a) The first line of input to each program will be either "black" or "white" (lower case) to indicate

Announcement

The third USENIX Computer Go Tournament will be held during the Summer 1986 USENIX conference in Atlanta, Georgia.

All interested parties are invited to submit programs.

The rules will be those established for the first USENIX Computer Go Tournament (listed below).

This event will be a "championship," i.e. the winner will be the "USENIX Computer Go Champion" until the next championship is held (most probably at the Summer 1987 conference).

Conference attendees may bring programs with them and get in touch with Peter Langston by noon on Tuesday, June 10th. People who are unable to attend the conference but would like to enter their programs can do so by sending a compilable source to the address below, (or by taking a chance and sending an "executable" file which may, or may not, function under last minute operating systems changes or machine changes, or ...).

The source code for the referee program to be used has been distributed through netnews *net.sources* and will be redistributed if interest warrants.

Comments, suggestions, programs, etc., should be sent via *uucp* to *bellcore!psl* or via U.S. Mail to:

;login:

- which color the program will be playing (and thereby whether the program plays first or second).
- b) The placement of a stone will be expressed as letter-number (e.g. "G7", note capitalization).
 - c) A pass will be expressed as "pass" (lower case).
 - d) The command "byo-romi" (lower case) means the time limit has been exceeded and all further moves must be generated within the 10 second time limit.
2. All output from the competing programs will be in the form of lines of characters sent to the standard output, terminated by a newline, and had better either be flushed after every line or be unbuffered to start with (e.g. "setbuf(stdout, 0);").
- a) The placement of a stone must be expressed as upper-case letter-number (e.g. "G12").

- b) A pass must be expressed as "pass" (lower case).

- c) Any output lines not beginning with "A" through "T" (excluding "I") or "pass" will be considered garbage and ignored.

- **"Bad" moves are a forfeit.** Any syntactically correct but semantically illegal move will be considered a forfeit. The three possibilities are: playing on a non-empty spot (occupied or off the board), *ko* violation, and suicide.

- **Play will end when both programs pass in sequence.** The programs may pass at any time, but once both pass concurrently, the game is over.

- **The decisions of the judge will be final.** A human judge will evaluate each game's results and may fill in missed *dame* or may judge a game incomplete if, in the judge's opinion, too much is unresolved. In general, Japanese rules will be used, (Nihon Kiin).

!!!NEWS FLASH!!!

4.3BSD WAS RELEASED

AND

WENT INTO THE MAIL

ON

FRIDAY, MAY 16



USENIX

Atlanta

THE PROFESSIONAL AND TECHNICAL UNIX* ASSOCIATION

USENIX Association

requests the honor of
your attendance at the

1986 Summer USENIX Technical Conference and Vendor Exhibition

Atlanta Hilton Hotel
Atlanta, Georgia
June 9 - 13, 1986

*UNIX is a trademark of AT&T Bell Laboratories.



INTRODUCTION

The Summer 1986 USENIX Technical Conference and Exhibition scheduled for June 9 - 13, 1986 in Atlanta, Georgia is sponsored by the USENIX Association. The USENIX Conferences are dedicated to fostering the development and communication of research and technological information and ideas pertaining to UNIX and UNIX-related systems. The Atlanta Hilton has been selected as the Conference Headquarters.

This brochure contains all the important information about the Technical Conference and Exhibition, including:

- Tutorials
- Technical Sessions
- Vendor Exhibition
- Hotels - and how to make your reservations
- Special Airline fares - and who to call



SCHEDULE OF EVENTS

(All events are being held at the Atlanta Hilton)

- **All-day Tutorials** Monday and Tuesday, June 9 & 10, 9am - 5pm
- **Vendor Exhibition** Tuesday, June 10, 12pm - 7pm
Wednesday, June 11, 10am - 5pm
Thursday, June 12, 10am - 5pm
- **Technical Sessions** Wednesday, June 11, 9am - 5:30pm
Thursday, June 12, 9am - 5:30pm
Friday, June 13, 9am - 3:30pm
- **Hotel Registration Deadline** May 10, 1986
- **Pre-registration Deadline** May 21, 1986
- **Conference Registration Desk Hours** Sunday, June 8, 4pm - 9pm
Monday, June 9, 7:30am - 8pm
Tuesday, June 10, 7:30am - 8pm
Wednesday, June 11, 7:30am - 4:30pm
Thursday, June 12, 7:30am - 1:30pm



THE SPONSOR

The USENIX Association is a non-profit organization of AT&T licensees, sublicensees, and other persons formed for the purpose of exchanging information and ideas about UNIX and UNIX-like operating systems and the C programming language. The Association sponsors technical conferences, an annual vendor exhibition, and produces a newsletter called "login:", and serves as coordinator of a software exchange for appropriately licensed members.



CONFERENCE COMMITTEE

The organizing committee for the Atlanta Conference and Exhibition consists of the following people:

PROGRAM CHAIR:	Mike O'Dell	Group L Corporation
PROGRAM COMMITTEE:	John Chambers	MCC
	Mike Hawley	Lucasfilm Ltd.
	Sam Leffler	Lucasfilm Ltd.
	Jim McKie	Bell Communications Research
	Dennis Ritchie	AT&T Bell Laboratories
	Spencer Thomas	Univ. of Utah, Computer Science Department
TUTORIAL COORDINATOR:	Michael Tilson	Human Computing Resources Corp.
USENIX MEETING PLANNER:	Judith DesHarnais	
VENDOR EXHIBITION MANAGER:	John Donnelly	

CONFERENCE HOST: Medical Systems Development Corporation



VENDOR EXHIBITION

The Summer 1986 USENIX Vendor Exhibition scheduled for June 10-12, 1986 at the Atlanta Hilton is the USENIX Association's annual technical exhibition of UNIX hardware and software systems. The primary intent of this exhibition is to provide vendors an opportunity to display advanced technology relevant to the UNIX technical community. Participating vendors are encouraged to concentrate on showing technical advancements at this exhibition. Vendors wishing to participate should contact:

John Donnelly
USENIX Exhibit Office
Oak Bay Building
4750 Table Mesa Drive
Boulder, CO 80303
(303) 499-2600

The USENIX Association is once again sponsoring a demonstration of inter-networking among different vendors. The exhibit hall will be cabled to allow a local area network between participating vendors.



TUTORIALS AND TECHNICAL CONFERENCE REGISTRATION FEES

Tutorial Fees

The tutorial registration fee includes the following:

- Admission to your tutorial selection(s)
- Copy of tutorial hand-out materials relevant to your session(s)
- USENIX binder, paper and pen
- Admission to Vendor Exhibition

PRE-REGISTRATION TUTORIAL FEES

One all-day tutorial selected - **\$150.00**

Two all-day tutorials selected - **250.00**

(add \$50 to pre-registration fee if your registration is postmarked after 5/21/86)

Technical Conference Registration Fees

Registration fees include the following:

- Admission to all technical sessions
- Copy of Conference Proceedings
- Admission to Vendor Exhibition
- Admission to USENIX Georgia Railroad Depot Reception

	<i>*Member</i>	<i>Non-Member</i>	<i>Student</i>
Pre-registration fee (up to 5/21/86)	\$125	\$140	\$60
On-site registration fee (after 5/21/86)	175	190	60

*The member rate applies to members of the USENIX Association

FOR PRE-REGISTRATION FEES, REGISTRATION FORMS MUST BE RECEIVED WITH FULL PAYMENT AND POSTMARKED NO LATER THAN MAY 21, 1986.

VISA, MASTERCARD AND AMERICAN EXPRESS ARE ACCEPTED!



IF YOU WISH TO JOIN THE



You may designate \$15.00 of your non-member registration fee to pay for the remainder of a 1986 (July-December) individual membership in the USENIX Association. Just check the special box on the Conference registration form requesting membership.



REFUND / CANCELLATION POLICY

If you must CANCEL, all refund requests must be in writing and postmarked no later than June 2, 1986. No cancellations can be taken over the telephone.

If you have registered but are unable to attend, you may call to substitute another person in your place.



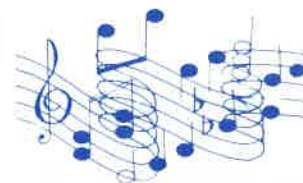
BIRDS OF A FEATHER SESSIONS (BOFS)

The USENIX Conference Office will schedule BOFS requests in advance of the conference, so that the times and locations can be included on the BOFS Bulletin Board in the registration area. On-site BOFS can still be scheduled and are encouraged. BOF sessions are available on Tuesday and Wednesday evenings.



CONFERENCE PROCEEDINGS

Conference Proceedings containing all papers submitted prior to the conference will be distributed at the conference during registration hours. One copy of the proceedings is included in the technical sessions registration fee. Additional copies of proceedings may be purchased at the conference registration desk or ordered after the conference from the USENIX Association office.



GEORGIA RAILROAD DEPOT RECEPTION

Join us for a great evening in an historic setting... Atlanta Style... Dixeland band, carnival games and Scarlett O'Hara.

Heavy hors d'oeuvres including fried chicken, whole pigs with carvers, barbecued beef sandwiches, corn on the cob and much more will be served.

The Depot is located just 5-10 minutes by bus from the Atlanta Hilton.

The reception and transportation are included in the technical conference registration fee. Additional reception tickets may be purchased at the Conference until noon on Thursday for \$35 each.



SPECIAL AIR FARES

Delta Airlines has been designated as the official air carrier for the USENIX Conference and Exhibition in Atlanta, Georgia, on June 9-13, 1986. However, the lowest available airfares at the time will be offered to each attendee.

Delta discounts are as follows:

- 40% discount — If ticketed at least 30 days from date of departure
- 35% discount — If ticketed at least 7 days from date of departure
- 30% discount — For all persons from Canada/Hawaii—if ticketed at least 7 days from date of departure

To ensure outstanding savings on airfare, call (toll free) 1-800-441-5081 (in California and outside the continental United States, call 714-756-0550). Please be sure to call between 8:00 am-5:30 pm Pacific Time. Ask for the USENIX Conference Agent.

Telex number is 495-1168.



HOTEL ACCOMMODATIONS

Special rates have been arranged for USENIX Conference attendees at the hotels listed below. A one night's deposit is required for each room reserved. Be sure to mention that you are attending the USENIX Conference.

HOTEL	ROOM RATES	
	SINGLE	DOUBLE
ATLANTA HILTON (Headquarters) 225 Courtland NE, Atlanta Complete the Atlanta Hilton reservation form and mail directly to the Hotel. You may call toll free to 1-800-Hiltons or directly at 404-659-2000 and ask for the front desk. <i>Upon USENIX approval, suites are available at the Atlanta Hilton.</i>	\$70	\$86
ATLANTA MARRIOTT MARQUIS 265 Peachtree Center Ave., Atlanta Call the Marriott Marquis at the toll free number 1-800-228-9290 or directly at 404-521-0000 and ask for reservations.	\$75	\$90

AIRPORT/HOTEL TRANSPORTATION

Atlanta Airport Shuttle provides transportation to the Atlanta Hilton and Atlanta Marriott Marquis every half hour, between the hours of 5am and midnight. Service is available right outside the public transportation exit from the baggage claim area. As of this printing, the cost is \$6 one way, \$11 round trip.

Taxi service is available at a FIXED FEE of \$13.50 for one person and \$14.50 for two persons.



USENIX TUTORIAL PROGRAM

UNIX Technology From the Experts

The USENIX Association is once again offering its well respected program of intensive UNIX tutorial sessions. These sessions are not "market overview" discussions—the tutorial sessions are taught by leading experts, are aimed at an audience of software professionals and technical managers, and should be immediately applicable to UNIX systems development and maintenance. This is your opportunity to learn from an expert at a reasonable cost and at a convenient time.

An expanded format will be used for the USENIX tutorial program in Atlanta. As a result of the tremendous demand at our January conference in Denver, Colorado, the most popular tutorials (Introduction to 4.2/4.3BSD Internals, Advanced Topics of 4.3BSD Internals, UNIX System V Internals, UNIX Device Drivers {4.2BSD}, and Local Networks) will be presented on Monday, June 9 AND Tuesday, June 10, 1986. By offering these full-day tutorials on both days we hope to allow more people to attend these continuously "sold out" tutorials.

The tutorial program has also expanded into new topic areas that we expect to be well attended by the UNIX community. Our popular traditional tutorials will also be offered. Attendance will be limited, and pre-registration is strongly advised. On-site registration will be allowed ONLY if space permits.

All tutorial sessions will be held at the Atlanta Hilton from 9am–5pm on Monday and Tuesday, June 9 and June 10, 1986. You must pick up your registration and tutorial hand-out materials at the Conference registration area before reporting to your tutorial session.

The Summer 1986 USENIX Tutorial Program is as follows:



INTRODUCTION TO 4.2/3BSD INTERNALS

Instructor: Thomas W. Doepfner, Jr.
Brown University



This tutorial is an introduction to 4.2BSD and 4.3BSD internals. It is geared to the programmer with a good knowledge of UNIX programming in C, but with little or no experience with UNIX internals. The course will cover process management, high-level I/O (including the file system), low-level I/O (i.e., device drivers), virtual memory, interprocess communication and networking. After taking the tutorial, the individual will have a basic knowledge of the structure of 4.2/3BSD and should be able to make his or her way through kernel code.

Thomas W. Doepfner Jr. received his Ph.D in Computer Science from Princeton University in 1977 and has been on the faculty at Brown University since 1976. He has lectured extensively on UNIX internals, over the past two years for the Institute for Advanced Professional Studies.

Important: You must be licensed for 4.2BSD source code (along with an appropriate UNIX source code license) in order to attend this tutorial. Please attach a copy of your institutional source license agreement or indicate your USENIX Institutional membership affiliation which we will use to verify your source license. Your signature is required on the registration form.



ADVANCED TOPICS ON 4.3BSD INTERNALS

Instructors: Marshall Kirk McKusick and Mike Karels
University of California, Berkeley



This tutorial is directed to systems programmers who have taken a course on 4.2BSD internals or who have had at least a year of experience working on the 4.2BSD kernel. The tutorial will cover the performance work done for 4.3BSD and will also discuss recent and planned changes to the kernel interfaces and facilities. The intent of the tutorial is to present a wide variety of material at a descriptive level. Presentations will emphasize code organization, data structures, and algorithms.

Kirk McKusick got his undergraduate degree in Electrical Engineering from Cornell University. His graduate work was done at the University of California, where he received Masters Degrees in Computer Science and Business Administration, and a Ph.D. in the area of programming languages. While at Berkeley he implemented the 4.2BSD fast file system and was involved in implementing the Berkeley Pascal system. He currently is the Research Computer Scientist at the Berkeley Computer Systems Research Group, continuing the development of future versions of Berkeley UNIX.

Mike Karels received his B.S. in Microbiology at the University of Notre Dame. While a graduate student at the University of California, he was the major contributor to the 2.9BSD release of the Berkeley Software Distribution for PDP-11s. He currently is the Principal Programmer at the Berkeley Computer Systems Research Group, continuing the development of future versions of Berkeley UNIX.

Important: You must be licensed for 4.2BSD source code (along with an appropriate UNIX source code license) in order to attend this tutorial. Please attach a copy of your institutional source license agreement or indicate your USENIX Institutional membership affiliation which we will use to verify your source license. Your signature is required on the registration form.



UNIX SYSTEM V INTERNALS

Instructors: Maury Bach and Steve Buroff
AT&T Information Systems



(Offered
Monday &
Tuesday)

This tutorial is a survey of the internal structure of AT&T's UNIX System V, and it is intended for people who maintain, modify or port UNIX systems. The tutorial will discuss traditional kernel concepts such as the file system, I/O subsystem, and process management, as well as new features in System V Release 3, such as demand paging, the file system switch, streams, remote file sharing and shared libraries. Attendees should have a good working knowledge of the UNIX system; basic kernel knowledge is recommended.

Maury Bach and Steve Buroff are Distinguished Members of the Technical Staff at AT&T Information Systems. Maury Bach has worked on multi-processor UNIX system development and streams. Steve Buroff has worked on multi-processor development and on paging virtual memory implementations. Bach has also taught a multi-week UNIX internals course within Bell Labs and is the author of the forthcoming book, *The Design of the UNIX Operating System*.

Important: You must be licensed for UNIX System V source code in order to attend this tutorial. Please attach a copy of your institutional source license agreement or indicate your USENIX Institutional membership affiliation which we will use to verify your source license. Your signature is required on the registration form.



UNIX DEVICE DRIVER DESIGN (4.2BSD)

Instructor: Daniel Klein
Consultant



(Offered
Monday &
Tuesday)

This course is designed for people who wish to become familiar with the fundamentals of designing UNIX device drivers. A knowledge of the major structures and internals of 4.2BSD UNIX is a desirable prerequisite for this tutorial, although a specific knowledge of the finer details is not required. This tutorial will cover the major aspects of driver design and implementation, and device integration. Both DMA and programmed I/O device drivers will be covered, as well as block and character (buffered and unbuffered) interfaces. We will outline the design and implementation of structured I/O devices (i.e., disk drives), and semi-structured devices (i.e., tape drives and serial communication links). This course will also discuss all aspects of adding a new device to the kernel (i.e., autoconfiguration, special files, device tables, and debugging). The intended audience for this course is systems programmers who will be actively engaged in the maintenance or design and implementation of UNIX device drivers. Although this course will be geared towards 4.2BSD, a comparison between the Berkeley and Bell Labs approaches will be offered. Users of System III or System V will therefore find this course to be informative.

Daniel Klein has been involved with UNIX since the original university distribution of Version 6 in 1976, including writing device drivers, utility programs, applications systems, and enhancements to the kernel. A graduate of Carnegie-Mellon University, Mr. Klein was manager of software systems at Mellon Institute for six years. He is presently engaged in teaching UNIX internals for the Institute for Advanced Professional Studies and developing an on-line educational system for UNIX, as well as developing a multi-processor simulation system.

Important: You must be licensed for 4.2BSD source code (along with an appropriate UNIX source code license) in order to attend this tutorial. Please attach a copy of your institutional source license agreement or indicate your USENIX Institutional membership affiliation which we will use to verify your source license. Your signature is required on the registration form.



LOCAL NETWORKS

Instructor: Bruce Borden
The Dana Group



(Offered
Monday &
Tuesday)

This tutorial presents an overview of local networking technology, with emphasis on UNIX implementations and futures. It will cover the ISO Open Systems Interconnection Model, the newly emerging ISO Protocol Standards, IP/TCP, XNS (SPP), NETBIOS, X.25, and other "standard" protocols. Various physical layers will be covered, including IEEE 802.X standards, Ethernet, PC-net, Pronet, Hyperchannel, etc. Emerging distributed file system implementations will be reviewed with an emphasis on their changing demands on protocol and media requirements. Finally, network performance will be addressed.

Bruce Borden is Vice President of Graphics for The Dana Group, developing a personal super computer. Prior to The Dana Group, Bruce was Director of Engineering for Silicon Graphics, founder of 3Com, developed the Excelan TCP/IP front-end protocol package, and authored the Rand MH mail handling system.



WRITING PORTABLE C PROGRAMS

Instructor: Tom Plum
Plum Hall, Inc.

(Monday
Only)

Today the C programming language is widely used to implement portable applications programs. But there are many pitfalls for the unwary, some obvious, but some very subtle. If you are not aware of the issues, it is easy to write programs that will not operate correctly in another hardware architecture, or another UNIX version, or another version of the C compiler. It then becomes expensive to move the application to a new machine. This course will teach you to recognize the trouble spots and avoid the pitfalls. You will learn to write truly machine and system independent code, and to protect yourself when this is not possible. This course is intended for experienced C application developers. If you are involved in the development of software which is to be used or distributed on a variety of systems, you should take this course.

Tom Plum is chairman of Plum Hall, Inc., a publishing and training firm specializing in the C language. He is author of two textbooks on C. Dr. Plum is also vice-chair of the ANSI X3J11 C Language Standards Committee.



THE NETWORK FILE SYSTEM (NFS)

Instructor: Mark Stein
Sun Microsystems, Inc.

(Monday
Only)

The Network File System (NFS) is a distributed file sharing service, designed for use in heterogeneous computer networks, which allows a client user to access files transparently across machine boundaries without regard to machine type or operating system used. Remote Procedure Call (RPC) and External Data Representation (XDR) are support protocols used by NFS and are available for use by other network services. With the growing support of NFS shown at UniForum 1986 in February (fifteen companies demonstrating interoperability on UNIX System V and 4.2 BSD, VMS, and MS-DOS), NFS is becoming an increasingly important force in UNIX networking. All NFS protocol specifications, together with RPC/XDR source code, are in the public domain. This tutorial is aimed at software professionals who would like to learn more about the technical details of NFS and related areas (RPC, XDR, and mount protocols, yellow pages data lookup service, transport and user interfaces, auxiliary services, and NFS administration). It will cover architectural, protocol and some implementation details of the various NFS components. In addition, we'll look at NFS's future directions.

Mark Stein is a Project Leader in the NFS Consulting group at Sun Microsystems. He is in touch with all aspects of NFS in the course of his work with the NFS vendor community, including conducting workshops and doing technical consulting. Mr. Stein taught the "UUCP, Mail, and News" tutorial at several previous USENIX conferences.



INTRODUCTION TO SYSTEM V UNIX SYSTEMS ADMINISTRATION

Instructors: Rebecca-Thomas and Rik Farrow
UNIX / WORLD

(Monday
Only)

This tutorial will cover the fundamentals of administering a UNIX System V system. The material is intended for administrators, not systems programmers. Attendees should understand the UNIX system at the user's level and be able to read simple Bourne shell scripts. We will discuss the file system – its constituent parts, how to create new file systems, how to interpret fsck reports, managing disk space, and backing up and restoring files. We'll also discuss system start-up and shutdown, controlling system access (includes disabling/enabling terminal lines, adding and removing accounts). Along the way, we will cover UNIX system security, the line printer spooling system, and UNIX communications (cu and uucp).

Dr. Rebecca Thomas is technical editor of *UNIX/WORLD* magazine and is author of two guide books on UNIX, *A User Guide to the UNIX System* and *Advanced Programmer's Guide to UNIX Systems V*. She is currently writing a book on UNIX system administration for Prentice-Hall.

Rik Farrow is a UNIX Systems consultant and writer. He has written UNIX installation manuals for three different computers, and numerous magazine articles. He and Dr. Thomas are currently writing a guide book on UNIX System Administration for Prentice-Hall.



SOFTWARE DEVELOPMENT USING C AND UNIX

Instructor: Rob Kolstad
Convex Computer Corporation

(Monday
Only)

This tutorial is for programmers who wish to use the full extent of software development tools available under UNIX. Programmers just getting into UNIX and C will be exposed to sets of tools which will allow them to exploit UNIX and – if they are used to another operating system – be able to do all those things they used to be able to do before. Proficient programmers might find their productivity increased as they exploit the available tools more effectively. One of the course's goals is to disseminate all those little "tricks" that experienced programmers know but which are written nowhere. The course is not a C tutorial, though it would make a fine adjunct for programmers who are learning C and need to learn the surrounding environment in order to make the smooth transition to the UNIX software engineering environment. The tutorial includes: comments on the UNIX philosophy, shell programming, writing and debugging C programs (including the preprocessor, lint, argument processing, debug schemes, conventions, symbolic debugging, and adb), optimization and profiling techniques, advanced problem-solving techniques (including plagiarism, configuration programs, file manipulation, and high level tools), project management techniques (including RCS), and hints on documentation. Extensive examples and explanations will accompany the text for this course.

Dr. Rob Kolstad has developed UNIX software for ten years – starting back in the days of Version 6. He currently manages both the operating system and computer operations groups at Convex Computer Corporation in Richardson, Texas. He received his Ph.D. in systems programming languages from the University of Illinois and is a frequent speaker at USENIX Conferences.



ADVANCED UNIX PROGRAMMING IN C

Instructor: Carol Meier
Emerging Technologies

(Monday
Only)

This tutorial will explore the details of processes in UNIX. While most of the concepts will apply to both AT&T and Berkeley UNIX, the implementation will be specific to System V UNIX. It is intended for programmers interested in understanding the following features of the UNIX system interface: how to use system calls, attributes of UNIX processes, process creation, environment manipulation, setting up pipes, implementing I/O redirection and background processing. A working knowledge of C language basics (basic data types, operators, expressions, statements, functions, simple arrays) is assumed. Advanced C topics necessary to understand the tutorial will be covered at the beginning.

(continued on page 8)

(continued from page 7)

After receiving her B.S. and M.S. degrees in Computer Science from the University of Pittsburgh, Carol Meier worked for Bell Laboratories as an applications programmer. Since 1983 she has worked as an independent consultant specializing in UNIX and C training and software development. She has presented dozens of public and on-site professional technical seminars throughout the United States and Canada. She has developed and currently teaches hands-on UNIX and C courses at the University of Colorado.



(Tuesday
Only)

LANGUAGE CONSTRUCTION TOOLS ON THE UNIX SYSTEM

Instructor: Stephen C. Johnson
The Dana Group

This tutorial is intended for C programmers who want to become familiar with the language development tools available on the UNIX system. The course will be directed towards application designers who may wish to use these tools to make front ends for their applications, rather than towards "traditional" compiler writing. Specific topics covered include: designing a language recognizer, the lex and yacc programs, symbol table issues, error reporting and recovery, strong type checking, and testing. Several in-class exercises will be given to lead the students through the construction of a simple front end.

Steve Johnson received his Ph.D. degree in pure mathematics from Columbia University in 1968. In 1967, he joined Bell Laboratories, Murray Hill, N.J., where he worked in psychometrics, computer music, and the computation center before joining the Computer Science Research Department. As a researcher, he worked on computer algebra, wrote the yacc parser generator, contributed to complexity theory and the theory of code generation and parsing, wrote the Portable C Compiler, and for several years, was involved in experimental VLSI design and Silicon Compilation. In 1983, he accepted a position with the AT&T computer line of business and was head of the Language Development Department in AT&T Information Systems. In 1986, he was named a Bell Laboratories Fellow. Dr. Johnson is currently Director of Programming Languages for The Dana Group in Sunnyvale, California.



(Tuesday
Only)

WINDOWING SYSTEM IMPLEMENTATIONS FOR BERKELEY UNIX

Instructor: David Rosenthal
Sun Microsystems, Inc.

The course covers window systems for Berkeley UNIX, examining four examples (Sun, Whitechapel, Andrew, and X) in considerable detail from the point of view of an application developer wishing to use one (or more). It discusses the techniques used to implement the systems, and their effects on application structure and performance, techniques for writing programs that port between systems, and the components that are missing from current systems. It also discusses the problems of porting window systems between displays and CPU's, and the problems of porting these systems to the System V environment.

David Rosenthal has been researching interactive graphics and user interfaces since 1968. He co-chaired the technical review of GKS, and was Associate Director of the Information Technology Center at Carnegie-Mellon University. He was one of the developers of the ITC's Andrew portable window system.



(Tuesday
Only)

INTRODUCTION TO 4BSD UNIX SYSTEMS ADMINISTRATION

Instructor: Ed Gould
mt Xinu

The basics of administering a 4BSD UNIX system will be covered. The tutorial will be oriented mainly towards Berkeley VAX UNIX systems. Topics covered will include system startup and shutdown, resource management, performance and tuning, the UNIX file system, and security, as well as others. The tutorial is designed for systems administrators, not for systems programmers. A rudimentary knowledge of UNIX is assumed.

Ed Gould has been working with UNIX since 1976. At the Computer Center at the University of California in Berkeley, he was involved with the management and administration of several systems that were used for general purpose timesharing for the campus. In 1983, along with Vance Vaughan and Bob Kridle, he founded mt Xinu, a company dedicated to the support and enhancement of technically advanced UNIX systems.



(Tuesday
Only)

MANAGING A LOCAL AREA NETWORK

Instructors: Evi Nemeth and Andy Rudoff
University of Colorado, Boulder

This tutorial is a summary of all the things we (and many others) have learned over the past couple of years in managing a growing local area network. It is intended for system administrators and others involved in planning, configuring, installing, and maintaining a networked UNIX facility. The course emphasizes 4.2/4.3BSD networks, yet includes issues that are global to all networks. Topics to be covered include:

- network overview (5%);
- buying and installing network hardware (15%);
- BSD network software installation (15%);
- non-BSD software/hardware installation (5%);

(continued on page 9)

(continued from page 8)

- experience with various vendor's products—discussion (5%);
- management issues, control, source code, users, resources, accounting (15%);
- tools to make these chores easier (10%);
- debugging—hardware and software (10%);
- writing programs that use the network (15%);
- mail (sendmail) and the network (5%);
- TCP/IP protocol demo (1%).

Evi Nemeth is on the Computer Science faculty at the University of Colorado and has built the University's Engineering Research Computing Facility from a single VAX 11/780 to its present complement of over 50 machines.

Andy Rudoff is a Computer Science student who has been involved with the systems work concerning this network's growth for the past 4 years. He is also working at ViaNetix, Inc. developing network systems software.



AIX ON THE RT PC

Instructors: Charlie Sauer & Larry Loucks
IBM

(Tuesday
Only)

This tutorial gives the software developer overview and background material designed to allow him or her to effectively develop software for the AIX operating system on the RT PC. Since the RT has a new hardware architecture, that architecture is first summarized. This includes a discussion of the RISC machine, 40-bit virtual memory, I/O system, floating point support, graphics support, and co-processing. The bulk of the tutorial is devoted to the software structure of AIX. This includes the relationship of AIX to standard UNIX System V, how AIX interfaces to the RT PC architecture, the Virtual Resource Manager, unique system services and their application, and the interfacing of device drivers to both AIX and the VRM. This tutorial presents a rare opportunity to learn about the structure of an important system as presented by senior architects of the system.

Dr. Sauer received his Ph.D. in computer sciences from the University of Texas at Austin in 1975. He is currently Manager of System Architecture for the IBM RT PC. Dr. Sauer has published three text books: *Computer System Performance Modeling*, co-authored by K.M. Chandy; *Simulation of Computer Communication Systems*, co-authored by E.A. MacNair; and *Elements of Practical Performance Modeling*, co-authored by E.A. MacNair.

Larry Loucks is a member of the IBM Senior Technical Staff and is the lead architect of the RT PC system. In his career at IBM since 1967, he has worked on a variety of products, including QTAM, TCAM, SNA, the 5520, and the RT PC.



USENIX TECHNICAL PROGRAM – WEDNESDAY, THURSDAY & FRIDAY

Preliminary Agenda – subject to change

WEDNESDAY, JUNE 11

9:00–10:30 WELCOME AND OPENING ANNOUNCEMENTS *Grand Ballroom*

KEYNOTE ADDRESS

Jon Bentley, AT&T Bell Laboratories—Jon Bentley writes the delightful column "Programming Pearls" in the Communications of the ACM.

10:30–11:00 BREAK

11:00–12:30 MUSIC *Grand Ballroom*

MIDI Music Software for UNIX

M. Hawley, The Droid Works

An Experiment in Music Generation

P. Langston, Bell Communications Research

12:30–2:00 LUNCH

2:00–3:30 NETWORKS 1 *Grand Ballroom*

Secure Networking in the Sun Environment

B. Taylor & D. Goldberg, Sun Microsystems

A Framework for Networking in System V

R. Israel, G. McGrath, D. Olander, AT&T Information Systems

OSI and TCP/IP Protocols on a System V System

J. Fenart, M. Fievet, C. Huitema, B. Martin, A. Remille, C. Vaysseix, INRIA

PERFORMANCE *Grand Salon*

Managing Development of Performance-Constrained UNIX-Based Software Systems

L. Perkins, Martin Marietta

A Multiuser Multiprocessor Benchmark to Compare UNIX Systems

P. Mills, NCR Corporation

A System Call Tracer for UNIX
R. Rodriguez, Digital Equipment Corporation

3:30-4:00 BREAK

4:00-5:30 OPERATING SYSTEMS 1 *Grand Ballroom*

A New Virtual Memory Implementation for UNIX
P. Clancy, E. Szynter, J. Crossland, Tektronix
Mach: A New Kernel Foundation for UNIX Development
M. Accetta, R. Baron, D. Golub, R. Rashid, A. Tevanian, M. Young
Carnegie-Mellon University
An Extensible I/O System
J. Rees, E. Shienbrood, P. Levine, Apollo Computer

TOOLS *Grand Salon*

The Care and Feeding of Relative Addresses
P. Honeyman & S. Bellovin, Princeton University
Dasde: A Document Annotation System for Distributed Environments
L. Cabrera & E. Mowat, IBM Almaden Research Center
Fully Automated System Administration
D. Nachbar, Bell Communications Research



THURSDAY, JUNE 12

9:00-10:30 NETWORKS 2 *Grand Ballroom*

Experiences Implementing BIND, A Distributed Name Server for the DARPA Internet.
J. Bloom & K. Dunlap, Univ. of CA, Computer Systems Res. Group
Network Performance and Management with 4.3BSD and IP/TCP
M. Karels, Univ. of CA., Computer Systems Res. Group
A Real-Time Electronic Conferencing System Based on Distributed UNIX
T. Suzuki, H. Taniguchi, H. Takada, NTT Electrical Comm. Lab.

REAL WORK 1 *Grand Salon*

How to Make Friends with Number-Crunchers
G. Dudek, M. Jenkin & H. Marcus, Univ. of Toronto
Kanjix UNIX: Yuniksu wa Nihongo o Hanasimasu (UNIX Speaks Japanese)
R. Jung & J. Kalash, Unisoft Systems
Tools for the Maintenance and Installation of a Large Software Distribution
D. Tilbrook & P. Place, Imperial Software Technology

10:30-11:00 BREAK

11:00-12:30 DISTRIBUTED FILE SYSTEMS 1 *Grand Ballroom*

Vnodes: An Architecture for Multiple File System Types in Sun UNIX
S. Kleiman, Sun Microsystems
Remote File Sharing Architectural Overview
A. Rifkin, AT&T Information Systems
A Generic File System Interface for UNIX
R. Hyde, M. Koehler, R. Rodriguez, Digital Equipment Corp., Ultrix Eng.

TEXT PROCESSING *Grand Salon*

Modelling Text as a Hierarchical Object
J. Waldo, Apollo Computer Inc.
STRIDE, An Emacs-based Document Processor
R. Droms & C. Kent, Purdue University
SMSCRIPT: An Interpreter for the Postscript Language Under UNIX
B. Borghi, S. Querel, D. de Raughlaudre, INRIA

12:30-2:00 LUNCH

2:00-3:30 DISTRIBUTED FILE SYSTEMS 2 *Grand Ballroom*

The Network File System Implemented on 4.3BSD
E. Gould, mt Xinu
Porting NFS to Systems V.2
M. Rosen & B. Fraser-Campbell, Lachman Associates
The Transparent Remote Filesystem
R. Hughes, Integrated Solutions, Inc.

LANGUAGE TECHNOLOGY *Grand Salon*

A Global Optimizer for Sun FORTRAN, C & PASCAL
V. Ghodssi, S. Muchnick, A. Wu, Sun Microsystems
Four Generations of the Portable C Compiler
D. Kristol, AT&T Information Systems
The Notifier
S. Evans, Sun Microsystems

3:30–4:00 BREAK

4:00–5:30 DISTRIBUTED FILE SYSTEMS 3 *Grand Ballroom*

Error Recovery in a Stateful Remote Filesystem
A. Atlas & P. Flinn, Masscomp
Distributed File Systems Panel

ELECTRONIC MAIL *Grand Salon*

Mail Routing Using Domain Names: An Informal Tour
C. Partridge, BBN Laboratories
AT&T Mail
D. DeJager, AT&T Information Systems
A Mail Delivery Agent for Eighth Edition UNIX
D. Hitz & P. Honeyman, Princeton University



FRIDAY, JUNE 13

9:00–10:30 OPERATING SYSTEMS 2 *Grand Ballroom*

Shared Libraries on the UNIX System
J. Arnold, AT&T Information Systems
Decreasing Realtime Process Dispatch Latency Through Kernel Preemption
D. Lennert, Hewlett Packard Company
MOS—Scaling Up UNIX
A. Barak, On Paradise, A. Shiloh, Hebrew Univ. of Jerusalem

USENET BOF *Grand Salon*

10:30–11:00 BREAK

11:00–12:30 WINDOWS *Grand Ballroom*

A Data-Flow Manager for an Interactive Programming Environment
P. Haerberli, Silicon Graphics
UWM: A User Interface for X Windows
M. Gancarz, Digital Equipment Corporation
Programming with Windows on the Major Workstations
S. Daniel & C. Rogers, Microelectronics Ctr. of NC

OPERATING SYSTEMS 3 *Grand Salon*

The Influence of Workload on Load Balancing Strategies
L. Cabrera, IBM Almaden Research Center
A. System V Compatible Implementation of 4.2BSD Job Control
D.C. Lennert, Hewlett Packard Company
UNIX as a Virtual Machine Environment
R. Genter, BBN Laboratories

12:30–2:00 LUNCH

2:00–3:30 REAL WORK 2 *Grand Ballroom*

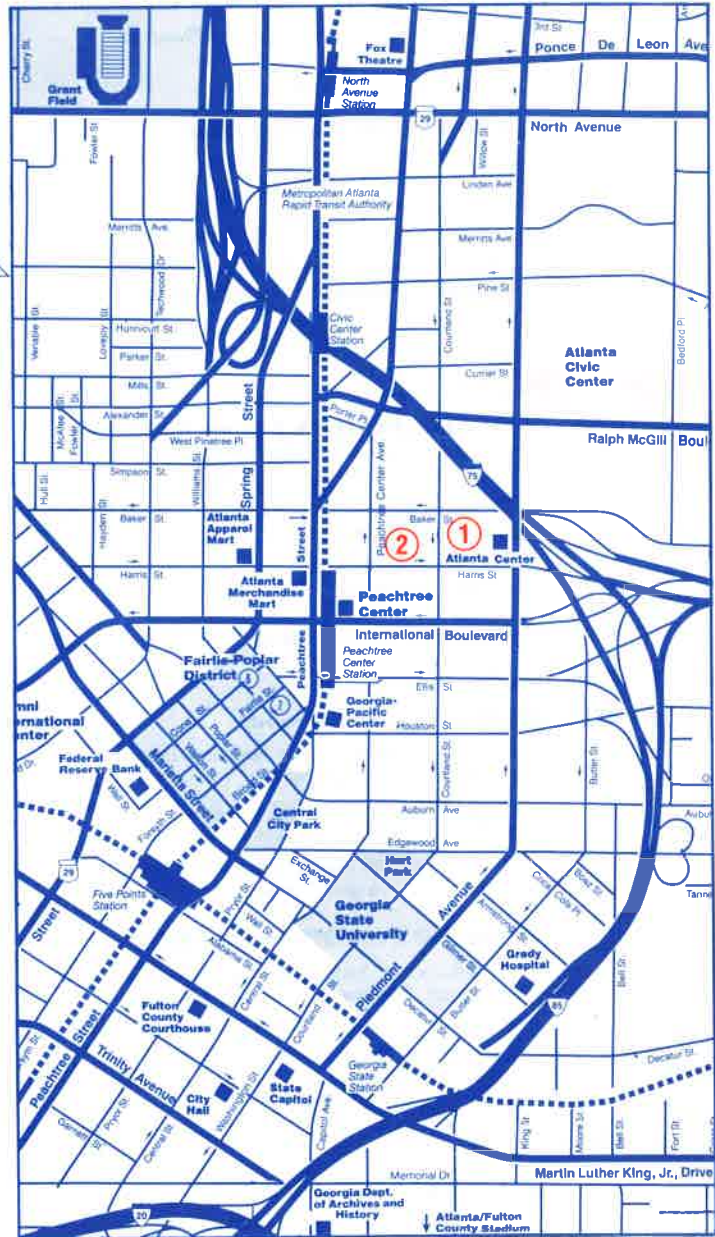
Porting to a Network of Discless Micros
W. Appelbe, D. Coleman, A. Fratkin, J. Hutchison, W. Savitch
Univ. of CA, San Diego, EECS Dept.
Experiences with a State-wide Distributed Computing System
B. Warren, T. Truscott & K. Moat, Research Triangle Institute
UNIX-based Distributed Printing in a Diverse Environment
W. Johnston, Lawrence Berkeley Laboratory

WIZARD'S PANEL *Grand Salon*

Metropolitan Atlanta



Downtown Atlanta



Atlanta Hilton and Towers ①
 Atlanta Marriott Marquis ②



For Additional Conference Information or Pre-registration Materials Contact:

USENIX CONFERENCE OFFICE

P.O. BOX 385, SUNSET BEACH, CA 90742 • (213) 592-1381/(213) 592-3243

;login:

HoneyDanBer UUCP – Bringing UNIX Systems into the Information Age

Bill Rieken
Middletown, NJ
ihnp4!opus!mtkam!wdr

Jim Webb
Bernardsville, NJ
ihnp4!opus!jrw

Part 1: Performance, Security, and Networking Facilities

Introduction

UUCP can be used to communicate with any of the 200,000+ UNIX[®] machines worldwide. Diagrams 1A and 1B show the backbone of the Usenet network of approximately 3,000 machines exchanging *netnews* articles every day. Any UNIX system can join this network, so as long as the world is connected by telephone lines, your personal computer becomes a powerful communications tool, thereby increasing its value to you as a useful information appliance.

UUCP makes this worldwide network possible. It allows us to *mail* our favorite recipes, wine lists, jokes, and flames to friends on other UNIX systems. It also lets us run commands on our friend's system, such as typesetting our resume on their laser printer, or forwarding our mail to Hong Kong via our friend's telephone line.

The original UUCP system was built by Mike Lesk at AT&T Bell Laboratories as part of a research project in 1976. It became so popular that a second, improved version was distributed with UNIX Version 7. This second version connected 80 machines in 1978. As the UUCP network grew, and better communication devices became available, problems with the second version became apparent. System administrators made local modifications to overcome some of the problems caused by the extraordinary use of UUCP. This proliferation of local versions of UUCP soon became another problem.

In 1983, Peter **Honeyman** (Princeton University), **David A. Nowitz** (AT&T Bell Laboratories), and **Brian E. Redman** (Bell Communications Research) rewrote UUCP. This third version became known as **HoneyDanBer UUCP** among UNIX enthusiasts, although AT&T calls it "Basic Networking Utilities" in the 3B2 manuals.

This article describes some of the major problems with the "old" Version 2 UUCP, and many of the improvements made by the "new" HoneyDanBer UUCP.

The list in Table 1 summarizes the HoneyDanBer improvements to "old" UUCP. These topics will be discussed in this article. We assume that you are already familiar with administration of the "old" (Version 2) UUCP.

Performance Enhancements

At 1200 baud it takes about 10 minutes to transfer a 60K file. Rather than make a user wait for completion, UUCP does its work in the background, freeing the user to do other things while the file transfer takes place. UUCP keeps track of each request in command (*C.*), data (*D.*), and execution (*X.*) files. These are kept in a "spool directory" (*/usr/spool/uucp*) until the file transfer is finished.

Usually UUCP tries to send the file(s) right away. However, the called system may be down, the telephone line could be busy, or all dialers may already be in use. In this case, UUCP notes the condition in a status file, and tries again later.

It's possible that a UUCP spool directory could have hundreds of queued requests on a heavily used system with many unreachable neighbors. A central netnews node will transfer about 2,000 netnews articles on an average day, and perhaps forward them to four or more satellite systems.

With 64 entries per directory block (System V), searching a huge spool directory could require several disk reads. If the system is also running many disk-intensive applications, the contention for disk I/O adds more delay to a spool directory search.

This article is © 1986 by Bill Rieken and Jim Webb – All Rights Reserved. It may be reproduced only by USENIX Association members for personal or in-house, non-commercial use. It may not be reproduced or distributed in any form for any other use without permission of the authors.

;login:

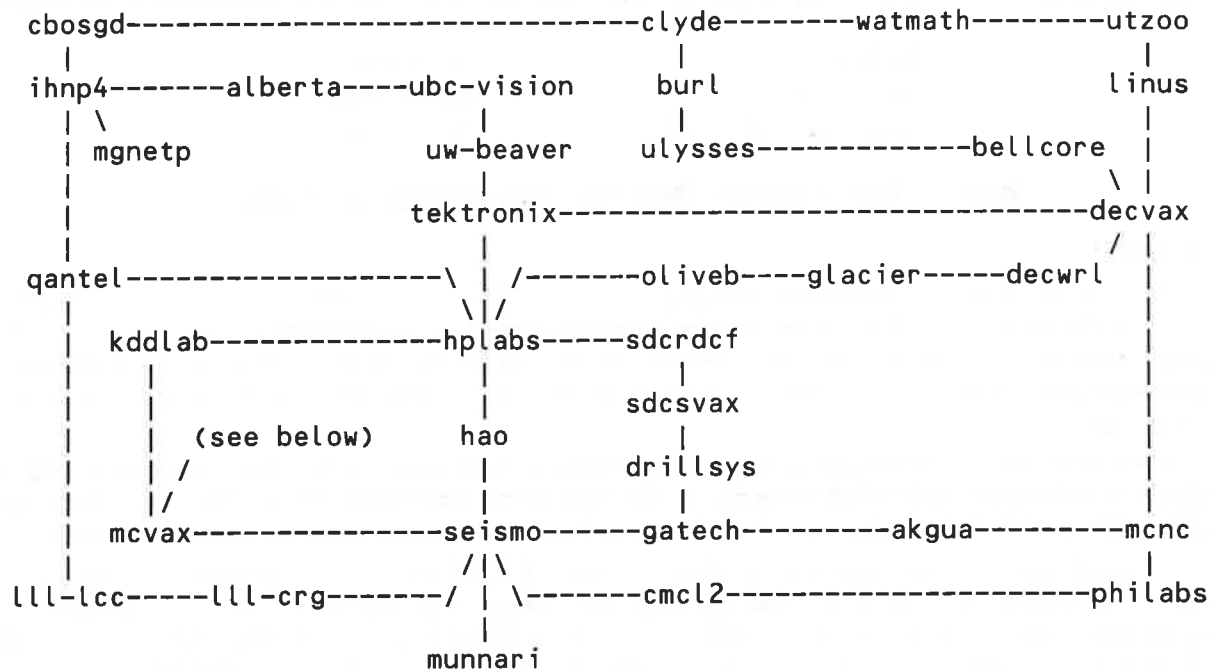


Diagram 1A: Usenet Backbone Nodes - USA - April, 1986

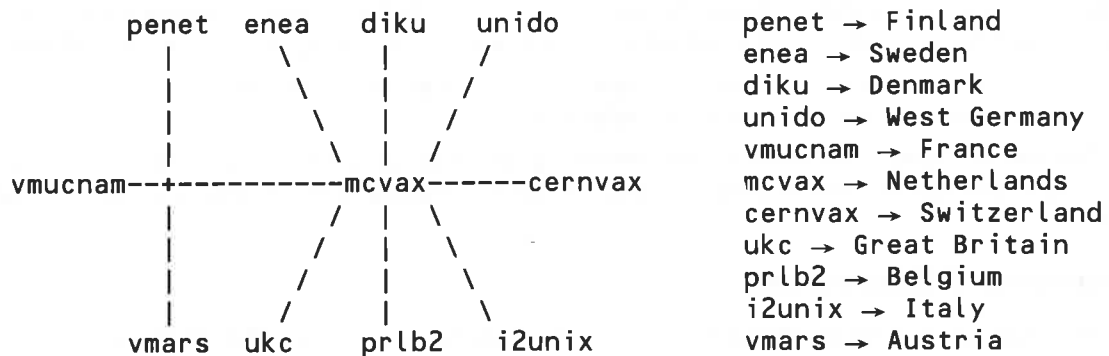


Diagram 1B: Usenet Backbone Nodes - Europe - April, 1986

How does this affect the UUCP system? Two talking systems are both executing the *copyin/copyout* command, *uucico*, to carry out the file transfer(s). The calling system is running as **MASTER**, and the called system is the **SLAVE**. The sequence of activity is as follows:

1. establish connection and agree on protocol
2. MASTER searches its spool directory and transfers files
3. roles are reversed (i.e., the called system is now MASTER)
4. step 2 is repeated
5. disconnect

After the calling system transfers all its files, the roles are reversed, and the calling system waits for the called system to search its spool directory for UUCP requests to the calling system.

;login:

But the calling system only waits for a "reasonable" amount of time, and then hangs up! If the called system cannot finish its directory search in a "reasonable" amount of time, requests queued for the calling system remain in the called system's spool directory, and the problem grows.

Worse, the *calling* system could take "too long" to search its *own* spool directory and the called system would hang up before *any* files were transferred! Whenever this happened, it usually required a system administrator to manually sort out the spool directory and "babysit" the file transfers until they were finished.

One of the first improvements to UUCP was suggested and implemented by Brian Redman – reorganize the spool directory to make it easier to find all the work for a given system. In HoneyDanBer, a subdirectory is created for each system with queued files. For example, UUCP requests queued for system "groucho" are kept in */usr/spool/uucp/groucho*, while */usr/spool/uucp/harpo* contains all the requests for system "harpo."

Note that this spool directory structure also makes life easier for the administrator, as the older Version 2 UUCP kept everything, lock files, data files, control files, etc., in the */usr/spool/uucp* directory, making it harder to find files during trouble-shooting sessions.

Security Features

Probably the best deterrent to outside intruders is the *uucico* command executed by UUCP whenever it logs in. Neither of the authors *admit* to being able to login and break out of this program to go exploring. This improvement was added in the second version of UUCP, as the original version gave the UUCP login a regular shell, which then executed *copyin* and *copyout* commands. Fine for a research environment among trusted systems, but far too permissive in today's network.

The second version provided additional protection by introducing several control files. The Version 2 UUCP administrator could specify in *USERFILE* which files on the local system could be accessed for transfer. The

UUCP Problems

single directory search
USERFILE, *L.cmds* security
Bell 801C/212A
only one "general" protocol
7-char site name
64-char UUCP path
losing files due to no space
multiple local versions
uninformative error msgs
unique configuration
UUCP litter
core program accounting
rmail, *rnews* handling
file forwarding quirks
no priority of remote cmds
Owner read-only files
poor documentation
Four decimal digit sequence no.
lock file hangs
55 minute retry, 10 hour max
specific UUCP functions
manual UUCP administration
few, cluttered admin files

HoneyDanBer Solutions

one directory per remote system
Permissions file flexibility
unlimited selection of *Dialers*
choice of LAN and WAN protocols
14-char site name
200-char UUCP path
read/write system call checks
standard HoneyDanBer
better error messages and err handling
parms.h, *makefile*, *uucheck*
uucleanup
uustat
general remote execution err handling
standard file forwarding scheme
"-g" option to prioritize remote cmds
UUCP makes a UUCP-owned copy in spool
comments in code; examples in manual
4 hex digit job no. / 3 hex digit seq no.
lock file error recovery
exponential retry for a week
generic lock, dial, X.25 packet driver
automated UUCP daemons and admin cmds
organized directory of admin files

Table 1: HoneyDanBer Improvements to Version 2 UUCP

;login:

L.cmds file specified which local commands could be executed via a *uux* command from a remote machine. Version 2 UUCP would accept files from machines listed in *ORIGFILE* for forwarding, but it would only forward to machines listed in *FWDFILE*. If, for example, HongKong appeared in *ORIGFILE* but not in *FWDFILE*, then our system would accept files from HongKong and forward them to any site listed in *FWDFILE*, but UUCP would not forward files to HongKong through our system.

These control facilities provided an improvement to the original UUCP, but their "all or nothing" options are limited compared to those provided in HoneyDanBer UUCP. For example, it might be desirable to let certain distinguished systems execute more commands than other systems, or give some systems access to more files than others. Also, it might be useful to specify read or write privileges independently on a per system basis. With HoneyDanBer, a software repository can be set up allowing any system to take files from a common directory, but only specified development machines are permitted to write new files into the common directory.

Nowitz designed and implemented HoneyDanBer's *Permissions* file to provide much finer control over who can do what on your system. **LOGNAME** entries specify permissions granted to calling systems, and **MACHINE** entries specify permissions in effect when your local system calls a remote one. Figure 1A shows an example of the flexibility given to you. Note that the *uucp* command must be allowed if you want other systems to forward *uuto* files through your machine. There will be a quiz on this in part 2 when we talk about error handling.

LOGNAME=nuucp \	! anyone who logs in as nuucp
NOREAD=/etc:/usr/src \	! cannot read /etc nor /usr/src
READ=/tmp:/usr/spool \	! but can read /tmp and /usr/spool
WRITE=/tmp:/usr/spool \	! and put files in those directories
REQUEST=yes \	! can request file copyouts
SENDFILES=yes \	! queued files will be sent
COMMANDS=rmail:rnews:uucp:lp	! and can execute these commands.

Figure 1A: HoneyDanBer Permissions Flexibility

The January, 1986 issue of *UNIX*[®] *Review* has an entertaining interview of Peter Honeyman by Eric Allman, author of *sendmail*, the (slowwww...) mail distribution facility used by 4.2 and 4.3BSD. The concepts discussed are enlightening, and the teasing between the System V (keep it simple and open-ended) and BSD (make it complicated and hard-coded) protagonists is amusing. [Note: The authors don't vigorously participate in the BSD/System V religious wars, but we couldn't resist this opportunity to express our personal biases.]

Allman (an absolute domainist) asked Honeyman (a path routing advocate) how to handle two nodes with the same name. Bewildered by Allman's mutterings about "political problems," Honeyman obviously didn't give this question much serious thought, and quickly conceded that one of the nodes would have to change its name (probably just to keep the interview from drifting too far away from technical issues). We hope the next example sets the record straight.

Our group had a lot of fun picking the name **opus** for our brand-new 8600, and we did not want to change it. Figure 1B shows how we used HoneyDanBer's facilities to keep our system name, after we learned that a central site already had an *opus* in their *Systems (L.sys)* file.

LOGNAME=huucp \	! when someone logs in as huucp
MYNAME=hropus \	! they know us as hropus

Figure 1B: Two Nodes with the Same Name

Rather than call every other node in the network and ask them to change our system name, the **MYNAME** option allows us to assume an "alias" whenever the central node logs in as **huucp**.

What happens when we login to the central node? How will it be able to distinguish us from the other *opus*? By adding a "MACHINE=central" line to our *Permissions* file, as shown in Figure 1C.

;login:

```
MACHINE=central \  
MYNAME=hropus
```

```
! when we login to the central node  
! it will know us as hropus
```

Figure 1C: Two Nodes with the Same Name Revisited

Unfortunately, this naming flexibility can be used by merry-makers and evil-doers to masquerade as a trusted system, so take care.

To thwart such merry-makers, HoneyDanBer has a **VALIDATE** option that requires specified systems to use a fixed login.

```
LOGNAME=dare \  
VALIDATE=make:my:day
```

Figure 2: A Trio of Trusted Machines

The example shown in Figure 2 requires that machines "make," "my," and "day" login as "dare." UUCP will hang up if these machines login as any other user. That's all it does for you.

Most people initially think that **VALIDATE** works like a password, but in fact it does not. The manual is not very clear about this, so we did the following experiment: from **mtkam** we logged into opus as "dare"! HoneyDanBer UUCP lets *any* system login as "dare," *provided it gives the correct password*.

So how does **VALIDATE** help protect your system? Consider the following scenario: Sneaky Pete lurks in the hallways and hears that "day" has special access to Mrs. Fields cookie recipes on system **clint**. So Pete uses the "MYNAME=day" option to gain UUCP access to the recipes. The **VALIDATE** option locks Pete out, because "day" must login as "**dare**," not "**nuucp**." Next, Pete tricks Sweet Sue into *cat*ting clint's *Permissions* file, and finds out that "day" logs in as "dare." This is the end of the line for our villain, because Secure Sam, clint's system administrator, was smart enough not to tell Sue the "dare" password.

Therefore, for **VALIDATE** to have any impact, the password for "dare" must be different and kept secure from the distributed *Systems* file; otherwise, merry-making is trivial. As long as this password remains secure, the administrator is guaranteed that no one is masquerading as these machines. With this in mind, the administrator can safely grant additional privileges to these machines, as shown in Figure 3.

```
MACHINE=make:my:day \  
REQUEST=yes \  
READ=/ WRITE=/ \  
COMMANDS=ALL
```

Figure 3: Additional Privileges Available to Trusted Machines

Note that this example gives these trusted machines free access to your system.

If the Administrator is really paranoid about imposters, the **CALLBACK** option can be used.

```
LOGNAME=5sides CALLBACK=yes
```

Figure 4: The CALLBACK Option

;login:

The example in Figure 4 specifies that all systems logging in as "5sides" will be called back to confirm their identity. For example, if machine "navy" logs in as "5sides," it will be called back using the information stored in the local *Systems* file. So will "army," "airforce," and "marines" – if they login as "5sides." Therefore, for someone to masquerade as "navy" they would have to modify the **local** *Systems* file to dial a different phone number.

Just make sure that two machines do not have "CALLBACK=yes" set for each other...

Incidentally, this "callback facility" existed in Version 2 UUCP. Its purpose was to allow a machine to login on a slower (1200 baud) modem port, and be called back using a faster (9600 baud) direct-connect link. Hardwired connections were one-way-only in Version 2 UUCP, as we'll see in the next section.

For new administrators, the distributed *Permissions* file may already be set up to provide a reasonably secure environment (see Figure 5).

LOGNAME=nuucp	! initial/default <i>Permissions</i> file
The defaults are:	
SENDFILES=no	even though you <i>say</i> you are a friend we won't send any files in our spool directory until WE CALL YOU.
REQUEST=no	you can't take any files from us
PUBDIR=/usr/spool/uucppublic	is your only R/W directory
and	
COMMANDS=rmail	is the only command you can execute

Figure 5: Default Permissions file

If you have purchased a source license, PUBDIR and DEFAULTCMDS are defined in */usr/src/cmd/uucp/parms.h*, which can be edited to *make* a new UUCP with different defaults. Of course, any default may be overridden by altering the corresponding *Permissions* file entry. Another safe default in *parms.h* is **NOSTRANGERS** – only systems that are in your *Systems* file are allowed to login – calls from "strangers" are rejected.

These are well documented in AT&T's "Basic Networking Utilities Guide" (Select Code 305-432) which can be ordered by calling (800) 432-6600. We recommend that you get this guide if you are interested in learning more about the initial setup and compilation of HoneyDanBer UUCP.

Networking Facilities

Passive sites in the UUCP network never call any other system. They simply keep their UUCP requests in the spool area, and wait for some other system to login as *uucp*. This arrangement is nice for low-budget shops wishing to minimize their telephone bills, and not in any hurry to communicate with other systems. However, most systems initiate UUCP action by calling a remote system. This requires hardware capable of dialing a telephone number, and software to give a number to the device, and wait for a connection.

Version 2 UUCP gave you two choices for outgoing lines: a hardwired link or a telephone dialup line (see Figure 6). Simple, inexpensive, and adequate for its use in Bell Laboratories.

DIRECT:	RS232 port	← →	null modem	← →	RS232 port
DIAL:	RS232 port	← →	801 dialer	← →	212/103 modem

Figure 6: Version 2 UUCP Connection Choices

;login:

Whether you set up a hardwired (DIRECT) or a telephone (DIAL) connection on a port, in both cases the port could only be used for *outgoing* calls. To allow either end to start a UUCP transfer required *two* ports.

our computer (no *getty*) → (*getty-login*) other computer
our computer (*getty-login*) ← (no *getty*) other computer

Figure 7: Version 2 UUCP Port Configurations

Once the connection was made, of course, data could flow both ways.

HoneyDanBer overcame this Version 2 limitation by providing a *uugetty* process which allows you to use a single port for dialin/dialout use. On the 3B2, the *sysadm uucpmgmt* menu option **bidirectional** starts a *uugetty* process on the port for you. On other machines, the administrator must edit */etc/inittab* to start a *uugetty*.

As intelligent (auto-answer/dial) modems became affordable to small computer users, and local area networks became available in large organizations, Version 2 UUCP had to be arm-twisted to use the new communication equipment. Today, HoneyDanBer offers you a choice of several popular smart modems and LANs to use, and its modular design makes it easier to add future communications devices.

Many feel that this is the most powerful enhancement in the HoneyDanBer package. Consider what had to be done to use a Hayes smartmodem with Version 2 UUCP....

Switch 6 must be down (Carrier Detect always on) for UNIX to open the device to send dialer commands. Fine for outgoing traffic, but what happens when you try to use the modem port for incoming calls? The *getty* process sees Carrier Detect, begins a login sequence, times out, and starts another login sequence. Worse, if there is an otherwise idle system connected to the port, they start trading "login invalid" messages even faster than *init* respawns *getty*, bringing your system to a virtual halt.

So you flip switch 6 up (Carrier Detect recognized) for the modem to interrupt only when someone tries to login. Great... now you can't use the Hayes for dialing. Unless you open the device with **FNDELAY** option to skip the wait for Carrier Detect... which requires a C program to do the open for you.

Or, you could do the following: Set switch 6 down, switch 3 up, and switch 4 down. The switch 3 and 4 settings effectively make the Hayes modem "shut up" (i.e., line noise, commands and result codes are not echoed to the computer). Switch 6 is not programmable, but 3 and 4 are. The Hayes command to change switches 3 and 4 are ATE0Q1 (quiet) and ATE1Q0 (talking).

So put the line in Figure 8 in your *L.sys* file.

```
sys Any tty1 1200 tty1 "" ATE1Q0DT9,5551212 in: uucp word: secret
```

Figure 8: Using a Hayes Modem with Version 2 UUCP

Wonderful. Now figure all this out again for every other modem you hitch up to your computer. And add a separate line for each and every different port you might use to reach system "sys." Then do the same for all the other systems in your *L.sys* file! AND DON'T FORGET TO CHANGE ALL THOSE L.SYS LINES IF YOU EVER NEED TO SWAP THE MODEMS ON 2 PORTS!! Egads, was "old" UUCP really that messy? We hope you get the point.

HoneyDanBer hides all the device-dependent knowledge in a separate *Dialers* file. And that second, redundant tty1 in Version 2's *L.sys* file has been eliminated from HoneyDanBer's *Systems* file. (One of Mr. Bill's big boo-boo's was to put "DIR" in place of one of the duplicate "tty" entries – which seemed logical for a hard-wired link, but Version 2 UUCP didn't understand "DIR" in the *L.sys* file! It *silently* went on to the next *L.sys* entry, and used a 1200 baud dialup line, rather than the 9600 baud direct link. Users began wondering why UUCP was *always* taking the slower connection? Sure, Mr. Bill eventually figured out what was going on and fixed it, and, of course, the friendly users understood how tricky UNIX can be... but all were secretly hoping for either a new system administrator or something like HoneyDanBer to make life easier for mere mortals.)

```
;login:
```

Systems file entry:

```

sys Any ACU 1200 9=555-1212 in: uucp word: secret
      |
      | Devices file index
      |
      | ACU tty1 - 1200 hayes
      | | | |
      | | modem 801
      | | port port
      | |
      | Dialers file index
      |
+-----+
|
|
+-----+----- expect -----+
| | | | |
hayes =, -, "" \dAT\r\c OK\r \EATDT0r\c CONNECT
|
+----- send -----+

```

Figure 9: HoneyDanBer UUCP Systems, Devices, and Dialers Files

Now, with HoneyDanBer UUCP, your *Systems* file entry will look like Figure 9, because it uses two other files, *Devices* and *Dialers* to try different ways to connect to "sys." The character pairs '=', and '-', are used to translate the "wait for dialtone" character (=) and "4-second pause" character (-) from a *Systems* telephone number into the corresponding ASCII character used by the Hayes modem (.).

The *Dialers* file has similar scripts for other popular smartmodems such as Penril, Ventel, Rixon, and Vadic, as well as some LANS such as Micom, Develcon, 3Bnet (Ethernet), and DATAKIT. Even X.25 and TCP/IP can be used by HoneyDanBer UUCP. The code that used to read like Figure 10A has been replaced by a function table

```

if (ACU)
    dial();                /* 801 ACU and 212/103 modem */
else
    connect();             /* hardwired connection */

```

Figure 10A: Version 2 UUCP Connect Logic

```

for each "sys" entry in Systems
    for each caller entry in Devices that matches the "sys" device
        if a dialer is not needed
            connect(NETWORK)
        else
            for each dialer in Dialers that can be used with this caller
                dial(DIALER)
    if not successful, try again later

```

Figure 10B: HoneyDanBer UUCP Connect Logic

;login:

similar to device switch tables in the kernel. The difference is that a single, known device driver is executed by the kernel, while HoneyDanBer tries several devices, as shown in Figure 10B. The priority of a device is given by the order in which it appears in a table.

You specify which modems/networks to include in this table and *make* UUCP. If the modem or network is not in this table, then a script can be added to the *Dialers* file. It's as simple as that.

This part of the rewrite was done by Peter Honeyman, when he was working for AT&T Bell Laboratories. If you ever have to add a new device to UUCP, you will certainly appreciate the extensibility of his design and the readability of his code!

Part One Summary

In this first article, we have given a brief overview of the major enhancements made by HoneyDanBer UUCP. The spool directory is now a tree with one directory for each remote site. Remote machines are not allowed to converse with your machine unless their name appears in your *Systems* file. You can specify the commands and files available to a remote system by name. The *Permissions* file is clearer and easier to understand than Version 2's *USERFILE*. The format of direct line connections in the *Systems* file is more natural, and new fields were added to the *Devices* file to index a variety of device protocol scripts in a *Dialers* file. A new program called *uugetty* is included so that a single port can be used for both dialin (*getty*) or dialout (*uucico*).

While HoneyDanBer UUCP is a very nicely done software package in its own right, it may be more noteworthy as a remarkable software engineering effort. Honeyman, Nowitz, and Redman were all working at AT&T Bell Laboratories in 1983 during this software development; however, they rarely saw each other at work! Almost all of their communication was via telephone and UUCP mail. Also, it is interesting to note how their different environments influenced the part of this UUCP-rewrite for which they accepted major responsibility. Nowitz was working on a project that had high security requirements, and thus did most of the *Permissions* file enhancements. Redman was using a machine with a large volume of netnews traffic and therefore was highly motivated to work on the Spool Directory/Performance areas. Honeyman's "path-routing advocacy" led him to work on the Local Area Network interface so that external UUCP mail addresses could be relative to a central "gateway" node (e.g., *allegro*), which would then forward mail to a recipient on a different local site machine known to the central site machine. He calls this "transitive closure of the name space," or something like that.

In Part 2 of this article, we will discuss error handling improvements, administrative aids, and user-level enhancements provided by HoneyDanBer UUCP.

;login:

Author Biographies[†]

Bill Rieken and Jim Webb are two casual heros who enjoy their adventures in the wild and crazy world of UNIX system administration. They founded “.sh consulting” in 1984, specializing in UNIX system training and consulting for several clients worldwide, including AT&T Bell Laboratories. Both have considerable expertise in several UNIX systems running on a variety of computers. One of them types faster than the other, who parties harder.

Acknowledgements

The authors would like to thank Peter Honeyman, David Nowitz, and Brian Redman for taking time out of their busy schedules to review this paper. Although they were probably bored to tears reading about code they had written three years ago, their comments were very kind and helpful to us.

A special thanks is owed to Rich Mayer, who never read a UNIX manual in his life but types very fast, for helping us “idiot proof” some of our examples. His motto regarding UNIX manuals is: “Don’t read the words. Just look at the pictures and examples. Get on a terminal and TRY IT!”

Others who helped are Edmond Gong, Wilton Chen, and Sam Torres, who nervously let us experiment with HoneyDanBer on their machines. Edmond provided the most help, although we won’t tell him how. We’d also like to thank Dr. Rick Welsch, whose careful critique of our gonzo journalism convinced us to abandon all hopes of ever writing for the National Academy of Sciences.

Finally, we’d like to thank Dr. Rebecca Thomas who introduced some semblance of technical writing into this paper.

Without the helpful cooperation of these people, the authors would probably still be hiding underneath a device driver somewhere in the South Pacific, sipping pinã coladas, and arguing about whether HoneyDanBer error messages are *really* any better than the old ones.

References

1. *AT&T 3B2 Computer UNIX System V Release 2.0 Basic Networking Utilities Guide*, (Select Code 305-432) Issue 1, October 1984.
2. *UNIX System V Administrator Guide – DEC Processors*, (Select Code 307-101) Issue 1, December 1983.
3. Version 2 and HoneyDanBer UUCP source code: `/usr/src/cmd/uucp`, `/usr/src/cmd/cu.c`, `/usr/src/cmd/ct.c`.
4. “An Interview with Peter Honeyman,” *UNIX Review*, January, 1986.
5. “A Dial-Up Network of UNIX Systems,” D. A. Nowitz and M. E. Lesk, and “UUCP Implementation Description,” D. A. Nowitz, *UNIX Programmer’s Manual, Volume 2*, Holt, Rinehart and Winston.
6. “Sendmail Revisited,” Eric Allman and Miriam Amos, *Summer 1985 USENIX Conference Proceedings*.
7. “Loving UUCP, or How I Spent My Summer Vacation” and “Experimental Implementation of UUCP – Security Aspects,” Honeyman, Nowitz, and Redman, “Mapping the UUCP Network,” Rob Kolstad and Karen Summers-Horton, *1984 UniForum Conference Proceedings*.
8. “Netnews via Satellite: A Progress Report,” Lauren Weinstein, “UUCP and Netmail,” Karen Summers-Horton, Mark Horton, Peter Honeyman, Pat Parseghian, Michael O’Brian, and Raymond Essick, *Winter 1985 USENIX Conference Proceedings*.
9. “UUCP Mapping Project,” *Summer 1984 USENIX Conference Proceedings*.

[†] The following is how the authors describe themselves. –Ed.

Future Meetings

USENIX 1986 Summer Conference and Exhibition June 9-13, 1986, Atlanta, Georgia

The USENIX 1986 Summer Conference and Exhibition will be held in Atlanta on June 9-13, 1986. There will be a conference, tutorials, vendor exhibits, and a Go Tournament. Please see the information elsewhere in this issue of *;login:*.

USENIX 1987 Winter Conference and UniForum 1987 - January 20-23, 1987, Washington, DC

The USENIX 1987 Winter Conference will be held at the Shoreham Hotel in Washington, DC, on January 20-23, 1987. It will be concurrent with UniForum 1987, which will be at the Washington Convention Center.

USENIX 1987 Summer Conference and Exhibition June 9-12, 1987, Phoenix, Arizona

The USENIX 1987 Summer Conference and Exhibition will be held in Phoenix on June 9-12, 1987. There will be a conference, tutorials, and vendor exhibits.

EUUG Fall 1986 Conference - September 22-25, 1986, Manchester, England

The next EUUG Fall Conference will be held in Manchester, England, September 22-25, 1986. It will take the form of a technical workshop dedicated to a specific topic related to UNIX systems. The subject of the workshop is "Distributed UNIX Systems."

Papers are solicited on the following topics:

- Design and implementation of distributed UNIX systems
- Mechanisms for distributing UNIX functions and services (message passing, remote procedure call, naming, etc.)
- Inter-process communication (IPC)
- Networking: protocols, addressing, etc.
- Remote file systems
- Loosely-coupled UNIX systems
- Multi-processor systems
- Parallelism
- Applications using distributed UNIX systems
- Heterogeneity and portability
- Distributed applications
- Software engineering and languages for distributed applications

- PC's in distributed UNIX environments
- Interconnecting UNIX to other systems
- Distributed computing, load sharing
- Fault tolerance
- Security
- Models for distributed UNIX systems
- etc.

Tutorials

It is planned to dedicate one day to advanced tutorials on subjects related to the Conference topics. Suggested topics for Conference tutorials include:

- Technology for distributed systems:
LAN (fiber optics), IPC's (Sockets, Streams)
- Networking: Protocols, ISO
- Remote File systems implementations (RFS, NFS)

Please let us know your interests, preferences and suggestions.

Paper Submission

Abstracts should be submitted to the EUUG Secretariat and to the Program Chair, by ordinary and electronic mail (if possible in *troff* -ms form). They should include the following information:

- Title of paper
- Author(s) name(s)
- Author(s) affiliation(s)
- Speaker name and affiliation
(if more than one author)
- Mail address
- Telephone number
- Fax and/or telex number
- Electronic mail address
- Special audio-visual requirements, if needed
- Text of abstract (in English): about 250 words

Deadlines

May 30, 1986: Abstract received by EUUG Secretariat and Program Chair

June 30, 1986: Notification of acceptance or rejection by Program Committee

August 30, 1986: Final paper received by EUUG Secretariat and Program Chair for publication in the Conference proceedings, (and free attendance to the Conference for the speaker).

;login:

Program Committee

Michel Gien (Chair)
PAA/TIM
CNET
38-40, Rue du Général Leclerc
92131 - Issy-les-Moulineaux France
Tel: +33 1 45 29 62 87
Fax: +33 1 45 29 60 38
E-mail: mcvax!vmucnam!mg

Hendrik-Jon Thomassen
AT Computing
Neil Todd
Department of Computer Science
University of Manchester

Secretariat

Mrs. Helen Gibbons
EUUG
Owles Hall
Buntingford, Herts., SG9 9PL Great Britain
Tel: +44 763 73039
E-mail: mcvax!euug

Local User Groups

The USENIX Association will support local user groups in the United States and Canada in the following ways:

- Assisting the formation of a local user group by doing an initial mailing for the group. This mailing may consist of a list supplied by the group, or may be derived from the USENIX membership list for the geographical area involved. At least one member of the organizing group must be a current member of the USENIX Association. Membership in the group must be open to the public.

- Publishing information on local user groups in ;login: giving the name, address, phone number, net address, time and location of meetings, etc. Announcements of special events are welcome; send them to the editor at the USENIX office.

Please contact the USENIX office if you need assistance in either of the above matters. Our current list of local groups follows.

In the **Boulder**, Colorado area a group meets about every two months at different sites for informal discussions.

Front Range Users Group
N.B.I., Inc.
P.O. Box 9001
Boulder, CO 80301

Steve Gaede (303) 444-5710
hao!nbires!gaede

Dallas / Fort Worth UNIX User's Group

Seny Systems, Inc.
5327 N. Central, #320
Dallas, TX 75205

Jim Hummel (214) 522-2324

In the **Washington, D.C.**, area there is an umbrella organization called Capitol Shell. It consists of commercial, government, educational, and individual UNIX enthusiasts. For information and a newsletter write:

Capitol Shell
8375 Leesburg Pike, #277
Vienna, VA 22180
seismo!cal-unix!capish

In the **New York City** area there is a non-profit organization for users and vendors of products and services for UNIX systems.

Unigroup of New York
G.P.O. Box 1931
New York, NY 10116

In **Minnesota** a group meets on the first Wednesday of each month. For information contact:

UNIX Users of Minnesota
Carolyn Downey (612) 934-1199

;login:

In the **Atlanta** area there is a group for people with interest in UNIX or UNIX-like systems:

Atlanta UNIX Users Group
P.O. Box 12241
Atlanta, GA 30355-2241

Marc Merlin (404) 255-2848
Mark Landry (404) 874 6037

In the **Seattle** area there is a group with over 150 members, a monthly newsletter and meetings the fourth Tuesday of each month.

Seattle / UNIX Group
P.O. 58852
Seattle, WA 98188

Irene Pasternack (206) FOR-UNIX
uw-beaver!tikal!ssc!slug

An informal group is starting in the **St. Louis** area:

St. Louis UNIX Users Group
Plus Five Computer Services
765 Westwood, 10A
Clayton, MO 63105

Eric Kiebler (314) 725-9492
ihnp4!plus5!sluug

In the **northern New England** area is a group that meets monthly at different sites. Contact one of the following for information:

Emily Bryant (603) 646-2999
Kiewit Computation Center
Dartmouth College
Hanover, NH 03755
decvax!dartvax!emilyb

David Marston (603) 883-3556
Daniel Webster College
University Drive
Nashua, NH 03063

A UNIX/C language users group has been formed in **Tulsa**. For current information on meetings, etc. contact:

Pete Rourke
\$USR
7340 East 25th Place
Tulsa, OK 74129

The **New Zealand** group provides an annual Workshop and Exhibition and a regular newsletter to its members.

New Zealand UNIX Systems User Group
P.O. Box 13056
University of Waikato
Hamilton, New Zealand

In the **San Antonio** area the San Antonio UNIX Users (SATUU) meet twice each month with the second Wednesday being a dinner meeting and the third Wednesday being a "roving" meeting at a user site.

San Antonio UNIX Users
7950 Floyd Curl Dr. #102
San Antonio, TX 78229-3955

William T. Blessum, M.D. (512) 692-0977
ihnp4!petro!bles!wtb

A new UNIX users group is starting in the **Coral Springs**, Florida, area. For information, contact:

S. Shaw McQuinn (305) 344-8686
8557 W. Sample Road
Coral Springs, FL 33065

USENIX Association
P.O. Box 7
El Cerrito, CA 94530

First Class Mail

FIRST CLASS MAIL
U.S. POSTAGE PAID
El Cerrito, CA 94530
Permit No. 87

Summer '86 Conference at Atlanta

Go Tournament at Atlanta

Elections Results

Call for Proposals for UUCP/Usenet

HoneyDanBer UUCP

Membership Survey Results

Change of Address Form

Please fill out and send the following form through the U.S. mail to the Association Office at the address above.

Name: _____ Member #: _____

OLD: _____

NEW: _____

Phone: _____

uucp: {decvax,ucbvax}! _____